NTNU – Trondheim
Norwegian University of
Science and Technology

A Comparison of Visual Evoked Potential (VEP)-Based Methods for the Low-Cost Emotiv EPOC Neuroheadset

Fredrik Tron Hvaring
Andreas H Ulltveit-Moe

Master of Science in Computer Science
Submission date: June 2014
Supervisor: Asbjørn Thomassen, IDI

Norwegian University of Science and Technology
Department of Computer and Information Science

Abstract

Brain computer interfaces (BCIs) enable interaction with computers through electrical brain signals recorded from the scalp through an electroencephalogram (EEG). These BCIs are characterized by expensive equipment and long setup times, which limits their commercial use. In this thesis, a BCI was implemented that uses the low-cost EEG acquisition device Emotiv EPOC and visual evoked potentials (VEPs), which are potentials in the EEG elicited by visual stimulus. A structured literature review was conducted to find which techniques are used in state of the art VEP-based BCIs. The four most promising techniques found through the literature review were implemented and tested using Emotiv EPOC. Three of these methods use steady state visual evoked potentials (SSVEPs), which are VEPs elicited by a periodic stimulus. Two of these methods use canonical correlation analysis (CCA) and one method uses power spectral density analysis as feature extraction techniques to detect frequency information in the recorded EEG. The last of the four methods uses code-modulated visual evoked potentials (c-VEPs), which are VEPs elicited by stimulus following a pseudorandom pattern. Experiments showed that a c-VEP implementation is not possible using Emotiv EPOC, due to a synchronization issue between the stimulus and the recorded EEG data. All three SSVEP techniques reached satisfactory results. One of the methods using CCA reached an average information transfer rate (ITR) of 32.92 bits/min, which is the highest reported ITR for any VEP-based BCI using Emotiv EPOC in an online setting.

Sammendrag

"Brain computer interfaces" (BCI-er) virkeliggjør interaksjon med datamaskiner gjennom opptak av elektriske hjernesignaler fra hodebunnen ved bruk av et elektroencefalogram (EEG). Disse BCI-ene har begrensede kommersielle bruksområder på grunn av dyrt utstyr og lang konfigureringstid. I denne avhandlingen ble et BCI implementert som tar i bruk lavkost EEG-opptaksenheten Emotiv EPOC og "visual evoked potentials" (VEP-er), som er en respons som forekommer i EEG og er framkalt av visuell stimuli. Et strukturert litteratursøk ble gjennomført for å finne teknikkene som er brukt i de beste VEP-baserte BCI-ene. De fire mest lovende teknikkene som ble funnet ved hjelp av litteratursøket ble implementert og testet ved bruk av Emotiv EPOC. Tre av disse metodene bruker "steady state visual evoked potentials" (SSVEP-er), som er VEP-er fremkalt av periodisk stimuli. To av disse metodene brukte "canonical correlation analysis" (CCA), mens en metode bruker "power spectral density analysis" for å hente ut relevant frekvensinformasjon fra de innspilte EEG-dataene. Den siste av de fire metodene bruker "code-modulated visual evoked potentials" (c-VEP-er), som er VEP-er fremkalt av stimuli som følger et kvasitilfeldig mønster. Eksperimenter viste at en c-VEP-implementasjon ikke er mulig ved bruk av Emotiv EPOC på grunn av et synkroniseringsproblem mellom den visuelle stimulien og de innspilte EEG-dataene. Alle de tre SSVEP-teknikkene oppnådde tilfredsstillende resultater. En av metodene som bruker CCA oppnådde en gjennomsnittlig "information transfer rate" (ITR) på 32.92 bit/min, hvilket er den høyeste rapporterte ITR-en for et VEP-basert BCI som bruker Emotiv EPOC i et miljø som er "online".

Preface

This thesis outlines the work we have done in our master's project, conducted at the Department of Computer and Information Science at the Norwegian University of Science and Technology (NTNU). Asbjørn Thomassen supervised this work, and we would like to thank him for invaluable support and advice. This thesis concludes our MSc degree in Computer Science.

Fredrik Hvaring and Andreas H. Ulltveit-Moe
Trondheim, June 12, 2014

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter introduces the work done in the thesis. Section 1.1 gives relevant background information as well as the motivation for the research. Section 1.2 presents the goal of this thesis and the research questions defined to reach the proposed goal. In Section 1.3, the research methods applied are described, and in Section 1.4, contributions of the work is presented. Finally, Section 1.5 gives an overview of the thesis structure.

1.1 Background and Motivation

Brain activity produces electrical signals. With hardware, these signals can be read from the surface of the scalp using electroencephalogram (EEG). Systems that use these electrical signals to infer intent are called brain-computer interfaces (BCIs). BCIs provide options for communication for people who suffer from severe neuromuscular conditions such as amyotrophic lateral sclerosis (ALS), brain stem stroke, cerebral palsy and spinal cord injury. These people are often unable to communicate using normal muscles such as speech or movement.

There are several paradigms within BCIs. Many paradigms use visual evoked potentials (VEPs) to infer the intent of the user using the system. VEPs are electrical signals that occur in the occipital and parietal lobes of the brain when a person is presented visual stimulus. Many VEP techniques require little to no user training, while maintaining a high performance in terms of speed and accuracy. This is what makes VEP-based BCIs popular.

One of the problems facing current BCIs is the cost of the equipment used. Medical grade EEG acquisition devices can have a price-tag in the tens of thousands of US dollars. These devices are cumbersome to use. They require a full head-covering cap with a high number of wires attached to them. It is often necessary to use a gel between the scalp and the sensors contained in the cap to record the EEG. The long setup time and steep costs of medical grade equipment makes these devices

unsuitable for commercial use.

Emotiv EPOC is a low-cost EEG acquisition device created for commercial use. It is a wireless headset with sensors attached to it, and requires minimal setup time. While Emotiv EPOC provides lower quality of the EEG than its medical grade counterparts, it does so at the fraction of a cost. VEP-based BCIs using medical grade equipment have performed well, and knowledge of how a low-cost headset performs in the same domain is valuable. Getting a BCI system to work with affordable hardware can be useful for those who suffer from neuromuscular conditions and cannot afford more expensive options. Few VEP-based BCIs have been used with Emotiv EPOC. The most used VEP paradigm, steady state visually evoked potentials (SSVEPs), has been applied with the headset [29], but not the more recently developed code-modulated VEP (c-VEP) method. It has not yet been demonstrated if the c-VEP paradigm is applicable to a less expensive EEG acquisition device.

Building a VEP-based BCI system is a challenging task and requires knowledge from several disciplines. A team building a VEP-based BCI can have experts within the fields of mathematics, statistics, signal processing, electronics, neuroscience, artificial intelligence and general computer science. In addition, the team has to know a great deal about techniques, schemes, terms and expressions within the field of VEPs and BCIs. The authors of this thesis are computer science students, with no previous knowledge about VEPs or BCIs. A considerably amount of reading had to be done in order to obtain the skills needed to approach the goal of this thesis.

1.2 Goals and Research Questions

In this section, the goal of the thesis is specified. In order to achieve the proposed goal, two research questions are defined. These questions are answered throughout this thesis. The goal of the thesis is as follows:

Goal To compare VEP-based BCI methods while using the low-cost EEG acquisition device Emotiv EPOC.

In this thesis, the words *methods* and *technique* are used for referring to detailed VEP implementations. The implementations can share characteristics and be an implementation of the same VEP paradigm, for instance SSVEP. The difference between the methods can be parameter settings, preprocessing of data, and the feature extraction and classification techniques used. The goal is referring to *methods*, thereby stating that the goal is to compare detailed VEP implementations. The second part of the goal states that the Emotiv EPOC headset is used. This is important, since part of the motivation for this thesis is to find methods that work well with a low-cost EEG acquisition device. The methods are compared to one another based on the information transfer rate (ITR), which is a performance metric used for BCIs that is calculated from the accuracy and detection time of

the method.

The following research questions drives the research towards the goal:

Research question 1 Which techniques can be used to classify the VEPs present in the EEG data?

To be able to compare techniques, it is essential to gather information about the existing VEP techniques in the field of BCI.

Research question 2 How is a system to compare VEP-based BCI methods implemented and configured?

Whereas research question 1 aims to gather theoretical information about VEP techniques, research question 2 addresses the problem of practically implementing a VEP based system. A VEP-based BCI typically consists of two parts; one part that presents stimuli to the user, and one part that processes the EEG response from the user and outputs a predicted command. Both of these parts need to be studied. The stimulus program can be configured in many ways, and a goal is to find the best way to configure this program.

1.3 Research Method

The research methods applied in this thesis are theoretical and design/experimental. Before any methods could be implemented, a structured literature review (SLR) was performed. The purpose of the SLR is to find information about what has been done within the field, what are the different methods used for VEP-based BCIs, and how a stimulus system is implemented.

The design/experimental phase implements a complete VEP-based BCI, including multiple VEP techniques and a visual stimulator. The different VEP techniques are then tested on multiple test subjects to assess which techniques work best on Emotiv EPOC.

1.4 Contributions

This thesis gives a comprehensive introduction to the field of VEPs. Through conducting an SLR, the state of the art VEP solutions are found and described in detail. The most promising methods are selected for implementation and testing in a complete VEP system using the low-cost Emotiv EPOC headset as an EEG acquisition device. The implementation reveals synchronization issues using the high performance VEP method, c-VEP, in combination with Emotiv EPOC. Tests show how the different VEP techniques compare. One of the methods reached an average ITR of 32.92 bits/min, which is the highest ITR reported to date in a VEP-based BCI using Emotiv EPOC in an online setting. The results also show that half of the test subjects performed significantly better with red colored visual

stimuli. This contradicts the findings by Cao et al. [5] which state that white color should elicit the strongest VEP response.

1.5 Outline

In this thesis, a system based on different VEP techniques is implemented. The techniques are implemented from both VEP paradigms, SSVEP and c-VEP, and are tested and compared with the use of Emotiv EPOC. The creation of the SSVEP system sheds more light on the performance of Emotiv EPOC, while implementing methods based on c-VEP brings new information to the field.

The structure of the thesis follows the twofold research method described in Section 1.3. Chapter 2 introduces the reader to background knowledge necessary to understand the later chapters. The chapter describes the analysis of EEG signals, the 10-20 system, the concept of BCIs, and the Emotiv EPOC headset used in this thesis. The end of Chapter 2 describes the structured literature review protocol used to perform an SLR. Chapter 3 presents the work found through the SLR, with explanation of central theories and techniques. Chapter 4 concerns the practical part of the thesis, with a description of the implemented VEP-system along with the justification for the choices made. In Chapter 5, the experiments and the results obtained are described. Finally, Chapter 6 gives an evaluation of the results and the thesis as a whole.

Chapter 2

Background Theory

This chapter gives an introduction to important background information that is necessary to understand the contents of Chapter 3. Section 2.1 gives an introduction to EEG, describing how it is used, how to interpret the signals and an explanation of the system used to specify electrode positions on the scalp. Section 2.2 gives an overview of what a BCI is, the two major paradigms, and how to evaluate BCIs. Next, Section 2.3 gives an overview of Emotiv EPOC, the EEG acquisition device used in this thesis, and how it compares to other medical grade EEG devices. Finally, Section 2.4 describes the SLR protocol and the research areas where the SLR was applied.

2.1 Electroencephalogram (EEG)

The electroencephalogram (EEG) is a record of oscillating brain electric potentials, recorded from electrodes attached to the human scalp [33]. The EEG shows how this electric activity changes over time, which can potentially give information about the state of the brain. Figure 2.1(a) shows an example of 10 seconds of EEG data.

EEG is traditionally used for medical purposes. The EEG can show patterns in electric activity that can be used to diagnose a number of conditions that affect the brain. One example is epilepsy, which is a condition that causes repeated brain seizures. EEG can help diagnose and manage this condition. EEG can also help in analyzing people with sleep disorders. In general, it can be used to identify areas of the brain that are not working properly. EEG is also applicable in other areas, such as analyzing head injuries, brain tumors, or measuring the brain function of individuals in a coma.

A disadvantage with EEG as a medical analysis tool is the limited spatial resolution. The rise of diagnostic methods such as computer tomography (CT) and magnetic resonance imaging (MRI) have made EEG less useful for analyzing brain

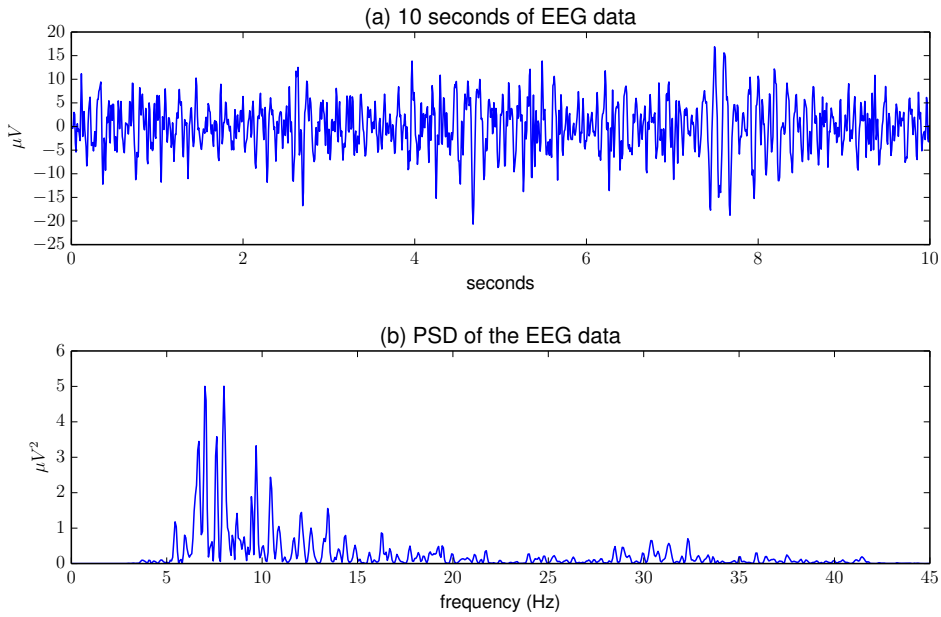

Figure 2.1: (a) shows an EEG recording with a duration of 10 seconds. (b) is the PSD calculated from the EEG data in (a).

disorders. The CT and MRI provide high-resolution images that give more precise information about the state of the brain. Although these methods give precise information about the brain at a given time, they do not have the temporal resolution to be able to track brain functions at a high rate. EEG, on the other hand, has a high temporal resolution. EEG acquisition devices can have sampling rates up to 5 kHz[1]. This means that EEG can be recorded with millisecond precision, which is not possible with CT or MRI. EEG, when recorded from the surface of the scalp, also has the advantage of rarely causing any side effects for the patient.

2.1.1 Interpreting EEG signals

An EEG measurement of the brain shows how the electric activity changes over time. How this signal is interpreted and manipulated depends on the purpose of the EEG recording. EEG can for instance be a valuable tool when analyzing the stages of sleep. In this case, it is interesting to see what frequencies are represented in the signal and how a given frequency band, ex. 0-4 Hz (the delta-band), behaves. The term *frequency band* is used to refer to a given frequency range in the signal. Examples of other bands are alpha (8-13 Hz), beta (12-30 Hz) and theta (4-7 Hz). Note that the words "band", "wave" and "rhythm" are used interchangeably throughout the EEG and BCI literature.

[1] `http://www.brainproducts.com/productdetails.php?id=5&tab=1`

One way to get frequency information from the EEG signal is to apply the Fourier Transform to transform the signal from the time-domain into the frequency domain. The Fourier transform converts a time-domain signal of infinite duration into a continuous spectrum composed of an infinite number of sinusoids [12]. It can represent any piecewise continuous function and minimizes the least-square error between the function and its representation. There are several common conventions for defining the Fourier transform. One of them is:

$$F(s) = \int_{-\infty}^{\infty} f(x)\mathrm{e}^{-2\pi isx}\,\mathrm{d}x \tag{2.1}$$

When x represents time (with the SI unit of seconds), the transform variable s represents frequency (in hertz). i is the imaginary unit, which satisfies the equation $i^2 = -1$.

The EEG signal does not contain a continuous mathematical function, which is what the general Fourier transform requires; an EEG acquisition device outputs discrete sampling points. To be able to transform the discrete samples into the frequency domain, the discrete Fourier transform (DFT) has to be used. The DFT of N uniformly sampled data points x_j (where $j = 0, ..., N-1$) is defined by

$$X_k = \sum_{j=0}^{N-1} x_j \mathrm{e}^{-2\pi ijk/N}, k \in \mathbb{Z} \tag{2.2}$$

Usually, the DFT is computed by an algorithm known as the fast Fourier transform (FFT). The running time for the FFT algorithm is $O(N \log N)$, making it much faster than the $O(N^2)$ running time of a naive DFT implementation. Various FFT algorithms exist, and one of the more popular algorithms is the Cooley–Tukey algorithm.

Analyzing EEG in the frequency domain

The frequency information is often analyzed in a frequency spectrum. In this spectrum, it can been seen how prominent a given frequency or frequency band is in the signal. Doing frequency analysis assumes that the signal is unchanged over short periods of time. In EEG analysis, the length of this interval, called window, varies with the task performed by the user, but is typically some seconds. In this interval, it is assumed that the user has the same state of mind giving the same type of EEG signals. The frequency spectrum can be plotted with the y-axis being the spectral density and the x-axis showing the frequencies. The usual unit of spectral density is u^2/Hz, where u represents the unit of the data in the time-domain. This is known as the power spectral density (PSD). EEG is normally given in μV. The PSD of an EEG signal would then be given in $\mu V^2/Hz$, as shown in Figure 2.1(b).

The frequency spectrum is two dimensional, showing the spectral density of each

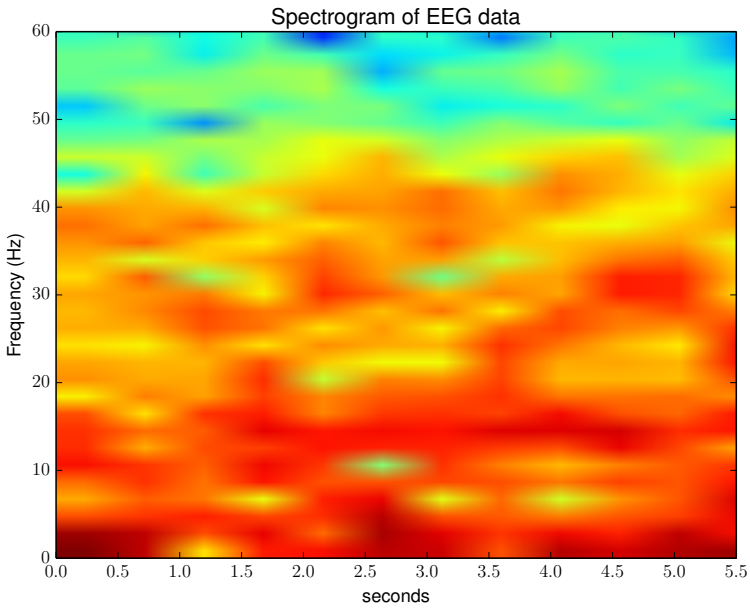

Figure 2.2: A spectrogram of EEG data for a duration of 5.5 seconds. Red color represent large magnitude, yellow represent medium magnitude, whereas blue color represent small magnitude. The spectrogram shows that most of the activity is below 40 Hz for the whole period.

frequency. It is possible to add time as another dimension, resulting in a plot called spectrogram (Figure 2.2). The plot is usually a 2D map with time on one axis and frequency on the other. The third dimension shows the power of a particular frequency at a given moment in time. In Figure 2.2, the power is represented by the different colors.

The literature will sometimes refer to the term band power. The *band power* to a specific frequency band is the total spectral density within the band. In other words, the band power will be a single number which says to what extent the frequencies in the band are represented in the signal. In the sleep analysis scenario, it would be beneficial to know the band power of the delta band.

Analyzing the EEG signal in the frequency domain assumes that interesting parts in the signal comes from repeated peaks in the time domain signal. In some applications, however, the interesting information does not lie in periodically repeated events, but events happening at a given time. For these applications the analysis of the EEG happens in the time domain.

2.1.2 The international 10-20 system

The 10-20 system, or the International 10-20 system, is an international standard that specifies electrode placements on the scalp. This system makes it easier to specify which positions on the scalp is used for a specific EEG experiment. The current electrode locations are shown in Figure 2.3. The 10-20 system increases the likelihood that an experiment can be reproduced.

Electrode locations consist of one or two letters followed by a number. The first letter identifies the lobe of the brain. The letters F, T, C, P, O stand for frontal, temporal, central, parietal, and occipital lobe, respectively (see Figure 2.4 for placement of these lobes). There is no central lobe. The "C" is only for ease of identification of the center positioned electrodes. In addition, the letter codes A, Pg and Fp identify the earlobes, nasopharyngeal and frontal polar positions respectively. There are also the letters AF, which is between Fp and F, and FC, which is between F and C.

The number specifies where on a given brain lobe the electrode is positioned. The positions on the right hemisphere of the brain are suffixed with even numbers (2,4,6,8) and the left hemisphere positions with odd numbers (1,3,5,7).

Guidelines [36] for the 10-20 system was released by the American Clinical Neurophysiology Society in 2006, and that standard is the one used in this thesis.

2.2 Brain-Computer Interfaces (BCI)

A brain-computer interface (BCI) is a system that creates an interface between brain activity and a computational device. EEG is the brain signal that is most widely used together with a BCI. An essential reason for this is the fact that the cortical synaptic actions generate electrical signals that change in the 10 to 100 millisecond range [33]. EEG and magnetoencephalography (MEG) are the only widely available technologies with sufficient temporal resolution to follow these fast dynamic changes. MEG systems, however, are bulky and expensive compared to hardware available for EEG [41].

A BCI is a continuous feedback loop; the user's intent is translated from EEG signals into a command that is sent to a device. The device provides feedback to the user of when a command is performed, and the user is free to focus on a new intent. The feedback loop is demonstrated in Figure 2.5.

Signal acquisition of EEG is performed by placing electrodes on the scalp. This signal is then digitized for use in a computational unit to perform signal processing. A main component of signal processing is to extract the relevant features from the signal that the classifier needs. EEG signals typically contain large amounts of noise, with the two main sources of noise being 1) imperfect recording methods

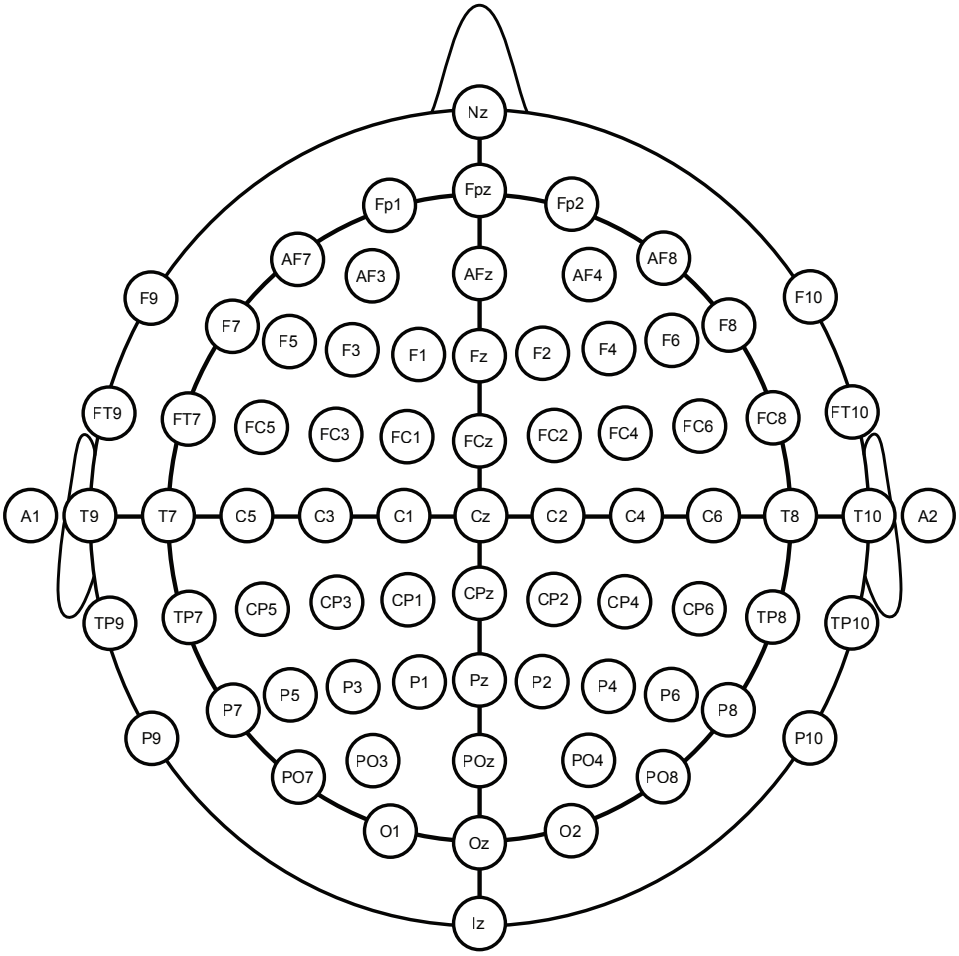

Figure 2.3: Electrode positions in the 10-20 system for EEG. By Marius 't Hart - `http://www.beteredingen.nl`. Used under CC BY: `http://creativecommons.org/licenses/by-sa/3.0/nl/deed.en_GB`.

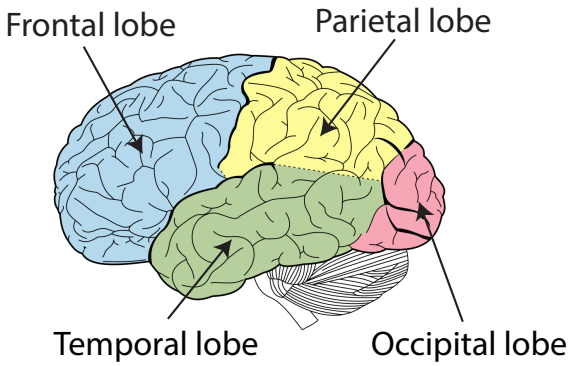

Figure 2.4: Lobes of the brain. The image is released under the public domain, adapted from `http://en.wikipedia.org/wiki/File:Lobes_of_the_brain_NL.svg`.

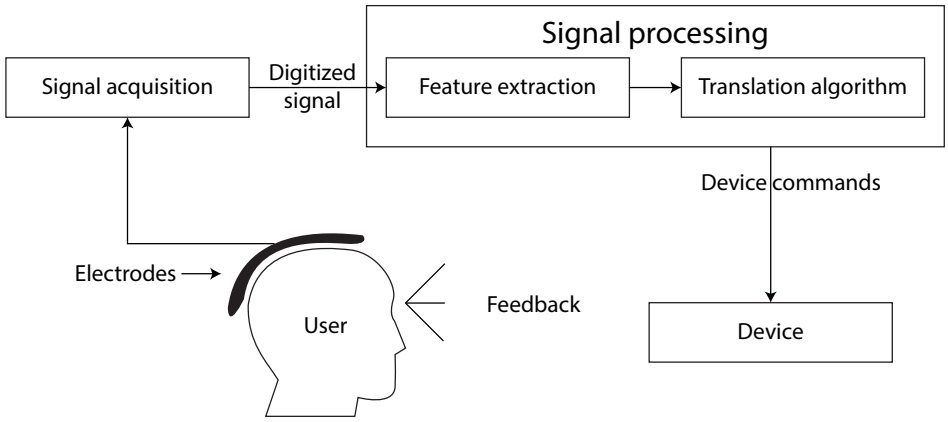

Figure 2.5: Overview of a BCI system.

and equipment, and 2) the body potential changing over time. It is common to filter the signal in an attempt to increase the signal-to-noise ratio (SNR). The SNR is defined as

$$SNR = \frac{P_{signal}}{P_{noise}} \tag{2.3}$$

where P denotes power. Feature extraction is then performed on the filtered signal. Examples of popular feature extraction methods are measuring the band power, calculating the power spectral density (PSD) or looking at time-frequency features [31].

When the signal has been filtered and the relevant features extracted, the data is ready to be classified. The purpose of classification is to determine the intent of the user of the system. If an intent is detected, a command is sent to the device connected to the system that performs the desired action. The device completes the feedback loop when it performs the action, giving the user feedback. Machine learning methods are the most prevalent classification methods within BCI research, with linear classifiers such as linear discriminant analysis (LDA) and support vector machines (SVMs) being the most popular algorithms [31].

Current day BCI research reflects two major paradigms: evoked potentials (EPs) and oscillatory features [32]. EPs are distinct waveforms that are phase-locked to an event, such as a visual stimuli. The other paradigm, oscillatory features, mostly revolves around spectral analysis such as looking at the power of different bands to determine intent or the affective state of the user.

2.2.1 Evoked potentials

The two most popular techniques that use EPs are (1) VEPs, and (2) P300, which is a component of an event-related potential (ERP).

An ERP is the measured brain response that is the direct result of a specific sensory, cognitive or motor event. ERPs can be classified according to the latency at which their components occur after stimulus presentation. ERPs with short latency typically occur at < 100ms after stimulus. These components are generated during the sensory stimulus processing stages in the brain, and they are named exogenous components because they are a direct response to an outside stimulus source. ERPs with long latency occur at > 100ms after stimulus, and represents the cortical processing stages. They are called endogenous components since they are less determined by the physical features of the stimulus. P300 is the component of the ERP elicited in the process of decision making, and as such is an endogenous component. It is called P300 because when recorded with EEG, it evokes a positive peak over the parietal lobe with a latency of around 300 ms after a decision has been made, as seen in Figure 2.6. This signal is present in every human and

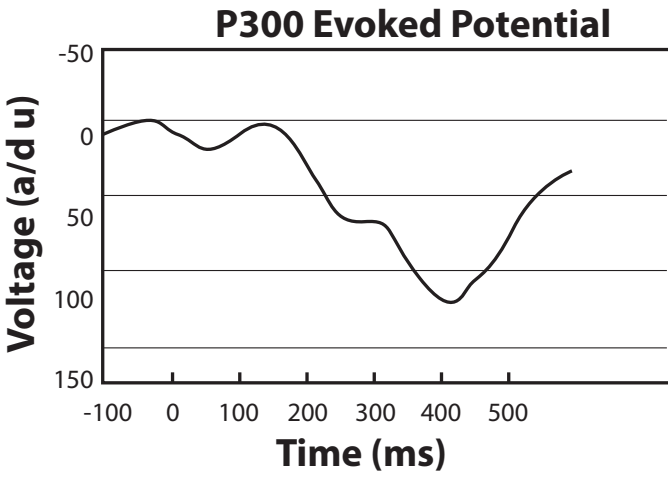

Figure 2.6: Example of a P300 signal. Adapted from McFarland and Wolpaw [32].

therefore requires little to no initial user training, making it a popular technique for BCIs.

Visual evoked potentials (VEPs)

VEPs are caused by sensory stimulation, and reflect the visual information processing mechanisms in the brain. This type of evoked potential can be seen in EEG recordings measured over the visual cortex, which is located in the occipital lobe in the back of the brain (see Figure 2.4). The visual cortex is the part of the cerebral cortex responsible for processing visual information. The purpose of a VEP-based BCI system is to determine what target the user of the system is visually fixated on by analyzing the concurrently recorded EEG. The targets themselves are each coded with a unique stimulus sequence (a flashing sequence), and by fixating on a target the user evokes a unique VEP pattern corresponding to the stimulus sequence of the target. This unique VEP pattern is then analyzed to determine which of the targets was fixated upon. These flashing sequences are most commonly presented on a monitor or by an external light source, such as an light-emitting diode (LED) [56]. Figure 2.7 shows a system diagram of a VEP-based BCI. This is a modified version of the system depicted in Figure 2.5, adding a device to present the visual stimuli.

VEP-based BCIs can be organized into three categories, based on the stimulus sequence approach: time modulated (time-modulated VEP (t-VEP)), frequency modulated (f-VEP or SSVEP), and pseudorandom code-modulated VEP (c-VEP) [1]. The idea behind t-VEP is for the flash sequences to be mutually independent of one another, for example by enforcing the flash sequences to be strictly non-overlapping in time. The stimuli is flashed briefly, and evoke a flash VEP (FVEP)

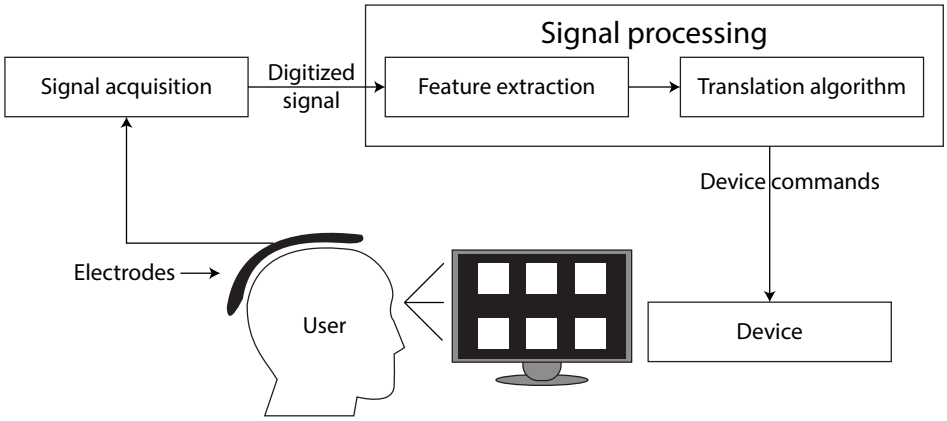

Figure 2.7: Example of a VEP-based BCI using a computer monitor. The user is presented with six targets, each following a different flickering sequence. The system finds which target the user gazed upon.

in the user. FVEPs have short latency and are considered a P2 (or P200) signal, evoking a positive peak with a latency of 200 ms after a flash [25]. In a t-VEP BCI, timing information is required for accurate classification. Accurate classification also requires averaging over many epochs (time periods), to enhance the FVEP from the fixation target and suppressing the FVEP from non-fixation target. Due to the requirement that flash sequences shall be mutually independent, t-VEP BCIs usually have low stimulus rates (< 4 Hz), resulting in low throughput compared to the other approaches [1].

In SSVEP-based BCIs, the targets flicker at different frequencies. Fixating on one of these targets creates a periodic sequence of evoked potentials with the same frequency as the target. It creates a response in both the fundamental frequency and its harmonics. The harmonics of a fundamental frequency, f, are the integer multiples of f. f is considered the first harmonic, $2f$ is the second harmonic, and so on. Spectral analysis is the most common classification technique for SSVEP-based BCIs [1]. SSVEP is also similar to P300 in that both methods require no user training.

c-VEP-based BCIs use pseudorandom stimulus sequences. A single pseudorandom stimulus sequence is shifted in time to create different stimulus sequences for multiple targets. Due to this shifting of the stimulus sequence, timing information is required to perform classification. Generally, a template matching method is used for classification [1]. This template T is obtained by performing a training session where the user is instructed to fixate on a target k in the system. The user has to fixate on the target for N stimulation cycles. The template is then obtained by averaging over the N cycles. To obtain the template T_i for a target

k_i, the template T is shifted in the time-domain according to the time-lag of the stimulus sequence between k and k_i. Classification can then be performed on a set of EEG data by calculating the correlation coefficients between the EEG data and all the target templates. The target with the highest correlation coefficient is selected by the classifier. A more thorough handling of c-VEP systems is given in Section 3.1.2.

2.2.2 Sensorimotor rhythms (SMRs)

A system based on sensorimotor rhytms (SMRs) is an example of a system that uses oscillatory features. The two main types of oscillations used for BCIs are the mu rhythm that oscillates in the range of 7-13 Hz, and the beta rhythm that oscillates in the range of 13-30 Hz [34]. The mu rhythm operates in the same frequency range as the alpha band (see Section 2.1), but measured over a different area of the brain. Both the mu and beta rhythm originate in the sensorimotor cortex of the brain, hence the name SMR. The sensorimotor cortex is an area of the cortex combining sensory and motor functions. Changes in these rhythms can be a result of sensory stimulation, motor behaviour, or mental imagery, and will result in either an amplitude suppression, event-related desynchronization (ERD), or an amplitude enhancement, event-related synchronization (ERS), in the EEG signal from this region of the brain.

Preparation or execution of a motor act results in a short-lasting ERD of the mu and beta rhythms [34], which is the idea used behind SMR BCIs. These ERDs can be read from the corresponding EEG signal and are used to infer intent from the user. Since an ERD can be triggered from an imagined motor act, SMRs are a popular field of study within the medical field for people with motor movement disabilities.

2.2.3 Evaluating BCI methods

Before comparing different BCI methods to one another, it is beneficial to take a look at how BCI methods are evaluated. The information transfer rate (ITR) is the most commonly applied metric to assess the overall performance of BCIs [53]. The method used for calculating ITR in BCIs was defined by Wolpaw et al. [49] as

$$B = \log_2 N + P \log_2 P + (1 - P) \log_2 \left(\frac{1 - P}{N - 1} \right) \tag{2.4}$$

where B is the ITR in bit rate (bit/symbol), N is the number of possible choices for symbols (targets), and P is the classifier accuracy. The bits/min, B_t, is calculated by

$$B_t = B \cdot \frac{60}{T} \tag{2.5}$$

where T is the detection speed in seconds/symbol.

A current problem with ITR is that different papers report ITR differently, or

even incorrectly. Some articles calculate the ITR based on offline analysis, which is considered to have little bearing on its online performance in a field setting. Yuan et al. [53] conducted a study on ITR in BCIs. The conclusions are summarized below.

There are four preconditions to using equation 2.4:

1. BCI systems are memory-less and stable discrete transmission channels.
2. All the output commands are equally likely to be selected.
3. The classification accuracy is the same for all target symbols.
4. The classification error is equally distributed among all the remaining symbols.

Most BCIs meet these preconditions in practice.

BCIs can be categorized by whether they are synchronous or asynchronous. In synchronous BCIs, the system provides the user with information about when to perform an action. These systems fulfill the preconditions listed above, and can use equation 2.4 for ITR estimation.

Asynchronous BCIs allow users to choose when they wish to control the BCI. This particular mechanic allows the user to remain in an idle state; any message sent during an idle period is a false positive, and should be avoided. This constraint can lead to the probability of choosing an idle state being different from the probability of selecting specific commands. Thus, asynchronous systems do not fulfill precondition (2), and can not use Equation 2.4.

Yuan et al. [53] have seven suggestions to standardize ITR calculations:

1. When reporting the ITR, N, P and T in Equation 2.4 and Equation 2.5 should be explicitly identified.
2. To ensure an accurate estimation of classifier accuracy, enough test trials are needed. Hence, when the ITR is reported, the number of test trials should also be reported.
3. Authors should include an ITR estimation that does not include error correction or other methods to increase effective throughput. If a system does employ error correction, authors should adequately describe the methods and results and, if desired, include a modified ITR as well.
4. To ensure that each input symbol is equally likely to be selected, BCIs should ideally be tested with randomly generated symbols from all N symbols.
5. When reporting ITRs, authors should explain all of the factors in the ITR calculation, such as whether t_1 is included in the calculations. t_1 is the time it takes the user to shift attention to a new target.

6. N should remain constant throughout the whole test.

7. Results should be presented from each subject tested, including individual ITRs and statistical results. If any data were rejected from further analysis, the amount of data and the reason(s) for rejection should be described. If results are presented from subject(s) who were exceptional, this fact should be noted.

In addition to the problem of articles reporting ITR differently as addressed by Yuan et al. [53] above, there are two other main criticisms against ITR as an evaluation method. The first criticism is that the ITR increases with the number of targets. If the detection speed (T in Equation 2.5) of a system remains unchanged regardless of the number of targets, then the highest bit rate will be obtained by choosing the value of N in Equation 2.4 for which B is the greatest. An example is given by Wolpaw et al. [49]: for a user with 90% accuracy when $N = 2$, 60% accuracy when $N = 4$ and 30% accuracy when $N = 16$, results in bit rates of 0.53, 0.40 and 0.38 respectively, and $N = 2$ would be the best choice. However, if the accuracies were 90%, 70% and 50%, the bit rates would be 0.53, 0.64 and 1.05, respectively. In the latter case, $N = 16$ would be the best choice.

The other criticism is regarding how ITR handles error rates. In many BCIs, when an incorrect selection is made it requires two additional selections to correct it: one selection for the "undo" command and one selection to perform the originally intended command. These new selections are also prone to the same error rate as the originally intended command. Townsend et al. [40] suggest a method called practical bit rate (PBR) that takes these extra selections into account. Given the error rate $E = (1 - P)$, where P is the classifier accuracy, and the number of commands the user has to input, S, the expected number of selections required to correctly input all commands is defined by:

$$S + 2(SE) + 2(2(SE)E) + 2(2(2(SE)E)E) + ... = S\sum_{i=0}^{\infty}(2E)^i = \frac{S}{1-2E} \quad (2.6)$$

when $E < 0.5$. When $E \geq 0.5$, the user would make errors at a faster rate than they would be able to correct them. $(1-2E)$ can also be written as $(2P-1)$. With this in mind, the formula for PBR in bits/min is as follows:

$$PBR = \begin{cases} B_t \cdot (2P-1) & P > 0.5 \\ 0 & P \leq 0.5 \end{cases} \quad (2.7)$$

where B_t is ITR given in bits/min as described in Equation 2.5. Speier et al. [37] suggest an alternative formula that uses $\log_2 N$ in place of ITR (B_t). Since both PBR and ITR include penalties for incorrect selections, the metric double counts errors, resulting in an overly conservative bit rate estimate. The alternative calculation is:

$$PBR = \begin{cases} \frac{60 \cdot \log_2 N(2P-1)}{T} & P > 0.5 \\ 0 & P \leq 0.5 \end{cases} \quad (2.8)$$

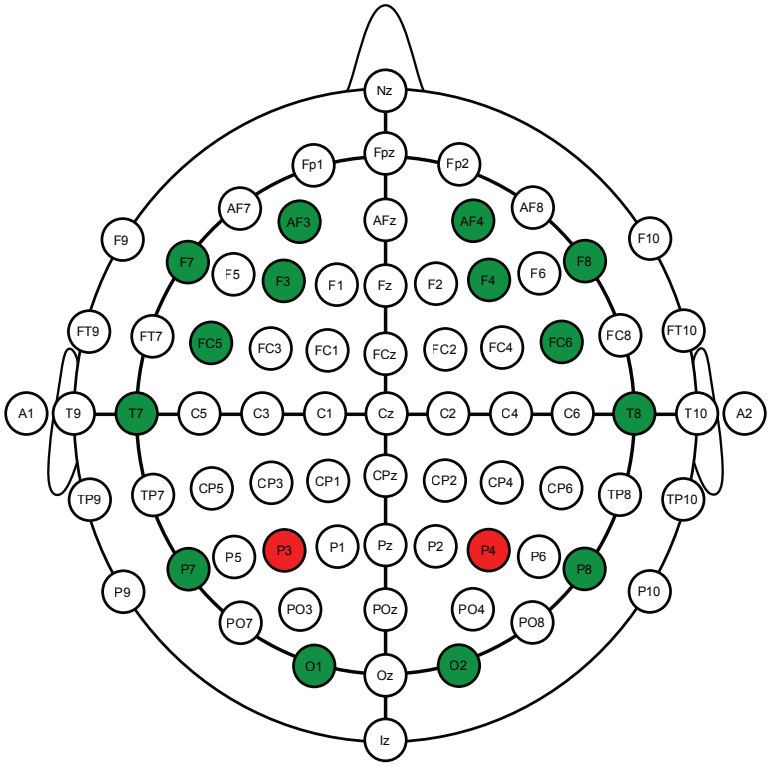

Figure 2.8: Sensor locations for Emotiv EPOC in the 10-20 international system. The green channels are the ones that provide EEG data. The red channels are reference (P3) and ground (P4).

where N is the number of possible selections and T is the average detection speed. The result is also given in bits/min.

While PBR can be a more correct performance metric, ITR is the most commonly applied metric, and the one that is used in this thesis. The VEP system in this thesis does not have any "undo" functionality, so PBR is less relevant.

2.3 Emotiv EPOC

Emotiv EPOC[2], shown in Figure 2.10, is a wireless, low-cost, multi-channel commercial EEG neuroheadset that records EEG data, and is the headset used in this thesis. Its features are shown in Table 2.1. The 16 sensor locations, including reference and ground sensors, are displayed in Figure 2.8. A reference sensor is a sensor whose EEG value is subtracted from all other sensors' EEG values. The

[2] `http://emotiv.com/store/sdk/209/`

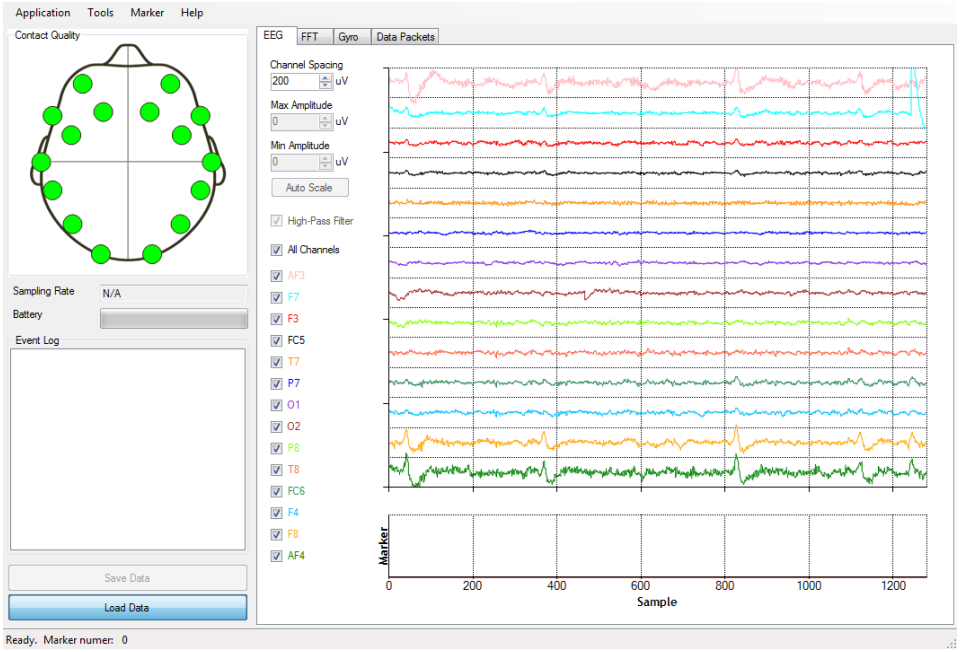

Figure 2.9: Screenshot from TestBench, software that is bundled with the Emotiv EPOC SDK.

O1 and O2 sensor placements, and to a lesser extent P7 and P8, cover the visual cortex, and can be used to record VEPs. These are the sensors that will be used in this thesis. On each of the Emotiv EPOC's sensors, felt pads are attached. To ensure good sensor contact, the felt pads have to be moistened by a saline solution prior to its use.

The Emotiv EPOC software development kit (SDK) comes with software, TestBench, that allows the user to check the contact quality of the sensors. Figure 2.9 shows a screenshot of TestBench. The contact quality ranges from black (no contact), to red (poor contact), yellow (good contact) and green (great contact). It also shows a graph of the EEG data in real-time.

2.3.1 Comparison to medical grade equipment

Stytsenko et al. [39] performed an evaluation of Emotiv EPOC, and compared it to a g.tec[3] device, which is a medical-grade EEG device. They concluded that the data provided by both systems are alike in general, but the signal has a better SNR in the medical system.

[3] `http://www.gtec.at/`

Figure 2.10: The Emotiv EPOC neuroheadset

Features	Emotiv EEG neuroheadset
Number of channels	14 (plus CMS/DRL references, P3/P4 locations)
Channel names (International 10-20 locations)	AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4
Sampling method	Sequential sampling. Single ADC
Sampling rate	128 samples per second (Hz) (2048 Hz internal)
Resolution	14 bits 1 LSB = 0.51 μV (16 bit ADC, 2 bits instrumental noise floor discarded)
Bandwidth	0.2 - 45 Hz, digital notch filters at 50 Hz and 60 Hz
Filtering	Built in digital 5th order Sinc filter
Dynamic range (input referred)	8400 μV (pp)
Coupling mode	AC coupled
Connectivity	Proprietary wireless, 2.4 GHz band
Power	LiPoly
Battery life (typical)	12 hours
Impedance Measurement	Real-time contact quality using patented system

Table 2.1: Emotiv EPOC features, taken from `http://emotiv.com/upload/manual/sdk/EPOCSpecifications.pdf`

Duvinage et al. [16] tested Emotiv EPOC and compared it to a product from Advanced Neuro Technology (ANT)[4], which is also a medical-grade EEG acquisition device. They reach the same conclusion as Stytsenko et al.; the SNR is worse for Emotiv EPOC than the medical-grade device. Furthermore, Duvinage et al. finds that Emotiv EPOC performs significantly worse, and it should not be used in health-related applications. However, for non-critical applications such as games, Emotiv EPOC should suit the customer's needs. While not performing as well as the medical device, it reaches satisfying results for an at least 40 times less expensive solution.

Emotiv EPOC also has more limited technical specifications compared to the more expensive solutions. As can be seen from 2.1, the headset has a sampling rate of 128 Hz and only has 14 channels. In comparison, medical equipment can have sampling rates of up to 5 kHz and 256 EEG channels[5].

2.4 Structured literature review protocol

A structured literature review is an effective method to gather information about a research area. The main advantage of conducting a structured literature review is the ability to systematically and objectively identify the existing solutions before the researcher(s) start working within a research field. A complete overview of the field will show where additional research is required, and where the focus should be to further strengthen the field. The review is documented in a formal manner, so that other researchers can reproduce or verify the review and understand what the given research is based upon. It can also help others who are looking for the same information, and avoids the necessity to duplicate the work. Furthermore, a structured literature review helps the researcher(s) in avoiding bias in the work and identify gaps of knowledge the researcher(s) might have.

When creating a VEP system there are two important areas that need to be addressed. The first is the large research area concerned with the algorithms and techniques used for processing EEG signals, recognizing VEP responses, and identifying the user's intent. The second area is the visual stimulus system used for provoking VEPs in the user. Providing the user with correct stimuli is crucial to making the system work. For this thesis, Emotiv EPOC is used, and investigating what has been done with the headset for VEP-based BCIs is a third area that needs to be addressed. These three areas formed the basis for the performed SLR. More formally they can be described as:

Pr1 Implementing techniques for VEP-based BCIs.

Pr2 Implementing a VEP-based BCI with Emotiv EPOC.

Pr3 Implementing a stimulus system for VEP-based BCIs.

[4] `http://www.ant-neuro.com/`

[5] `http://www.brainproducts.com/productdetails.php?id=5&tab=1`

The SLR in this thesis was carried out by the following steps

1. Defining problems and research questions.
2. Searching for relevant articles.
3. Filtering search results.
4. Collecting data and analyzing the findings.

A detailed description of these steps along with their results can be found in Appendix A. Among the articles found through the SLR, a set of articles were considered highly relevant. These articles form the basis for Chapter 3, where related work is presented along with an explanation of central theories and techniques within the field.

Chapter 3

Related work

In this chapter central theories and techniques from the articles found through the SLR (see Section 2.4) are presented. The first three sections present material regarding the three problem areas Pr1, Pr2, and Pr3 respectively. Section 3.4, contains a discussion of the material presented with a conclusion of the VEP techniques that are chosen to be implemented and tested in this thesis.

3.1 Pr1: Implementing techniques for VEP-based BCIs

As described in Section 2.2.1, there are three types of VEP categories; t-VEP, SSVEP and c-VEP. SSVEP is a VEP-based BCI in which the targets flicker with predetermined frequencies, whereas in c-VEP, the flickers follow predetermined patterns or sequences. VEP methods belonging to both of these paradigms were described in the articles found through the SLR. However, no articles were found that implemented a t-VEP system. This section is divided in two; one subsection presenting SSVEP techniques and the other presenting c-VEP techniques.

3.1.1 SSVEP methods

This section presents SSVEP techniques that were found through the SLR. The section will go through the following topics: PSD, canonical correlation analysis (CCA), adaptive time-window length, amplitude-modulated visual stimulation and minimum energy combination (MEC). PSD, CCA and MEC are methods used to detect SSVEP responses in the EEG, whereas the sections about adaptive time-window length and amplitude-modulated visual stimulation describe ways of making SSVEP systems.

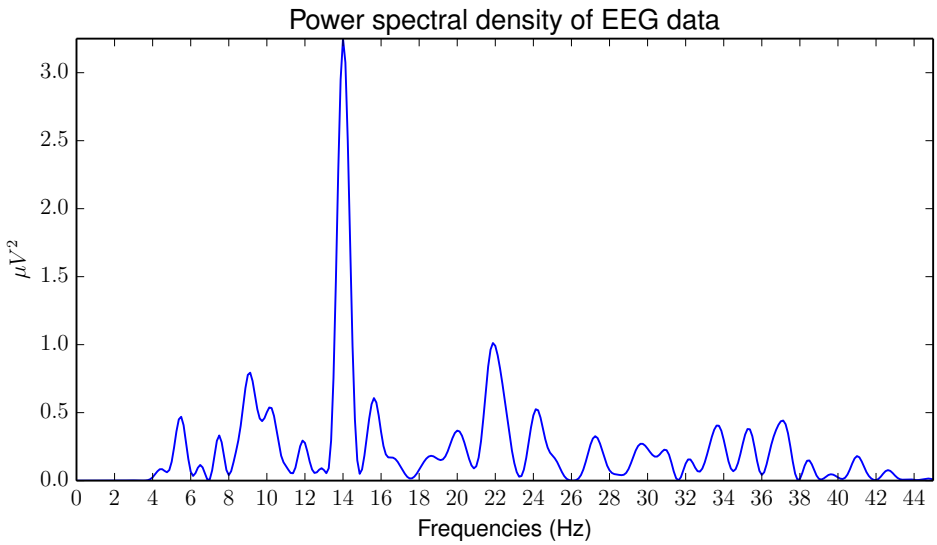

Figure 3.1: The PSD of the recorded EEG data of a user gazing at a target with a 14 Hz flicker frequency.

Power spectral density analysis (PSDA)

Until recent years, power spectral density analysis (PSDA) was the most widely used frequency detection method for SSVEP [28]. The PSD of a signal is a representation of the power of the frequencies in the given signal. Since SSVEPs are periodic responses in the EEG signal corresponding to which visual stimulus the user is gazing at, power spectral density analysis (PSDA) is a straight forward method to detect the frequency; if a user is gazing at an object that flickers at 14 Hz, the maximum amplitude in the PSD for an ideal, noise-free signal would be located at 14 Hz as well. The PSD can be estimated using the FFT. MathWorks, the company behind MATLAB, show how the PSD can be estimated using FFT[1]. The FFT of a signal returns both the amplitude and phase for the frequencies. The magnitude is given as a real number while the phase of the frequency is represented by a complex number. An FFT also returns the frequency information for both positive and negative frequencies. For the most part, EEG research is only interested in the real-valued part of the positive frequencies given by an FFT. Figure 3.1 shows the PSD of a user gazing at a 14 Hz flicker.

An FFT algorithm inherently assumes that the data given to it is infinitely repeated. Thus, if there is a height difference between the beginning and the end of the data, the FFT algorithm sees a step function once per cycle. This edge contributes to wideband noise in the frequency spectrum. Because of this, it is

[1] `http://www.mathworks.se/help/signal/ug/psd-estimate-using-fft.html`

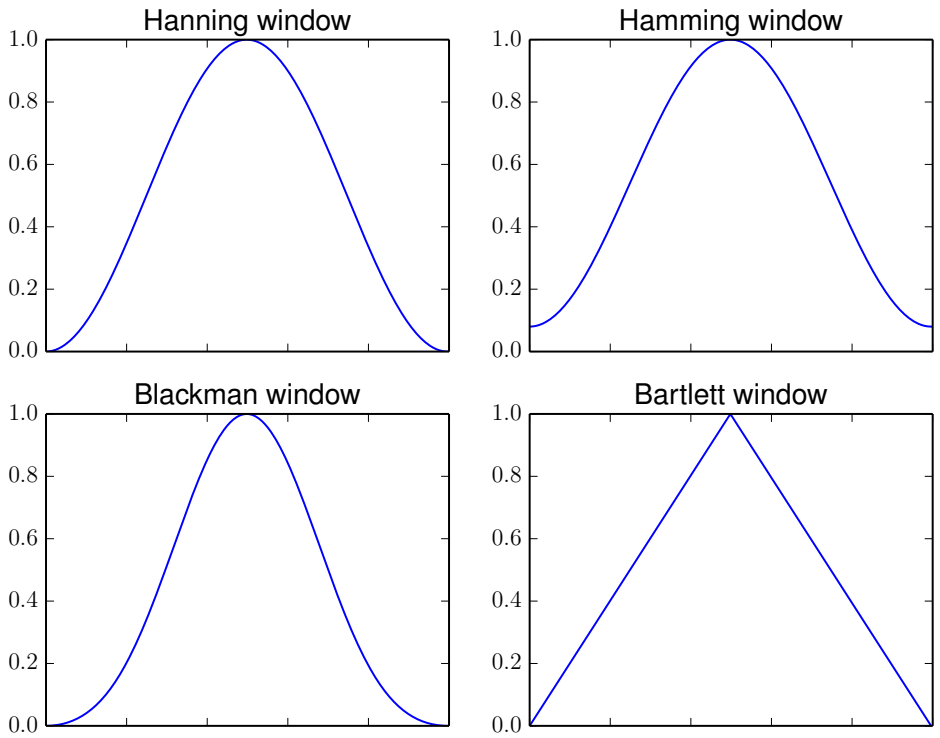

Figure 3.2: Four different windowing functions.

common to multiply the EEG signal with a windowing function of equal length before applying the FFT. A windowing function is a function that is zero-valued outside of some chosen interval. Figure 3.2 shows examples of four different windowing functions. A tapered windowing function is a windowing function that tapers from one in the middle to zero at both edges, and by multiplying the EEG signal by a tapered windowing function, the step function is no longer present. Figure 3.3 shows an example of a Hanning window applied to EEG data.

Before applying the FFT, it can also be beneficial to perform zero-padding on the EEG signal. Zero-padding means adding zeroes to the end of the signal. There are two reasons for performing zero-padding:

1. To set the signal length to a power of 2. FFT implementations are more efficient when calculating the FFT of a signal that has a length that is a power of 2.

2. To obtain a higher FFT resolution in the output.

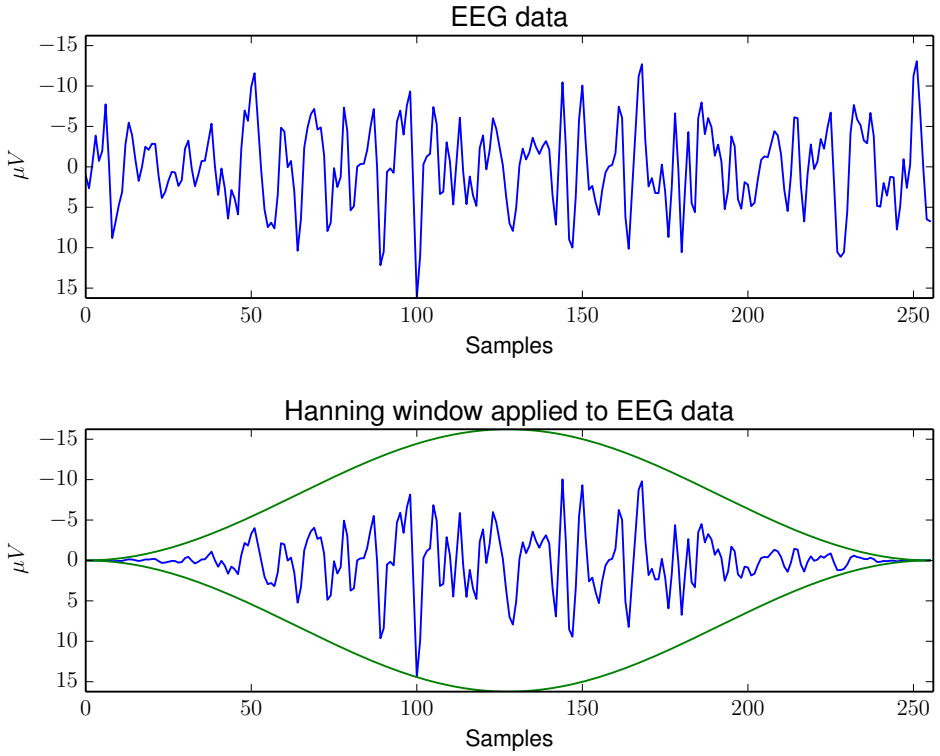

Figure 3.3: The effect of multiplying EEG data with a windowing function. The green line shows the maximum amplitude the data can have after applying the Hanning window.

The FFT resolution is the spacing between the frequencies in the output, and is given by the equation

$$\Delta R_{fft} = \frac{f_s}{N_{fft}} \tag{3.1}$$

where f_s is the sampling rate of the signal in seconds, and N_{fft} is the number of points in the signal with zero-padding. However, zero-padding will not provide more frequency information. Every signal has a frequency resolution which is the minimum spacing between two frequencies that can be resolved. The frequency resolution is determined solely by the acquisition time, and is given by the equation

$$\Delta R_f = \frac{1}{T} = \frac{f_s}{N} \tag{3.2}$$

where T is the time in seconds, f_s is the sampling rate of the signal in seconds, and N is the number of points in the signal without zero-padding [22]. To increase the frequency resolution, a longer signal is required. As an example, consider a signal where two frequencies of interest are spaced at 0.1 Hz. To be able to resolve both frequencies a frequency resolution of 0.1 Hz is required, and given that the frequency resolution is $\frac{1}{T}$ it would require 10 seconds of data. Finally, the frequencies returned by the FFT are multiples of the FFT resolution; if the frequency of interest is not a multiple of the FFT resolution, the amplitude of that frequency is split between the two closest frequencies.

There were two articles that passed all the criteria in the SLR that use PSDA. **Vilic et al. [42]** created a spelling system that uses dictionary support. The user of the system can choose whether to input a single letter, or auto complete a word from a dictionary based on the letters that have been typed. The experiments were carried out with an EEG acquisition device from g.tec[2] called g.USBamp with a sampling rate of 512 Hz. The signal was filtered by an analog bandpass from 5Hz to 30Hz, and data was collected from an electrode placed at the Oz location in the 10-20 international system, see Section 2.1.2. The visual stimuli, eight targets, were presented on a liquid crystal display (LCD) monitor with a refresh rate of 120 Hz. Classification is performed every two seconds on the EEG data, and two sets of data are examined during each iteration. The first set, $SData$, contains the two most recent seconds of data. The second set, $CData$, contains the three most recent sets of $SData$.

Autocorrelation is applied on $SData$ to reduce the noise. Autocorrelation is the cross-correlation of a signal with itself. It provides a measure of similarity between a signal and itself at different lags. Imagine a finite signal with n samples. The correlation between the signal and the signal at $lag(0)$ is calculated. Then the correlation between the signal and the signal at $lag(1)$ is calculated, where $lag(1)$ means the signal is shifted 1 sample to the right. This process is repeated until $lag(n-1)$, for a total of n correlations. The correlation is naturally highest at $lag(0)$. Autocorrelation can be used to find repeating patterns in a time-domain

[2] `http://www.gtec.at/`

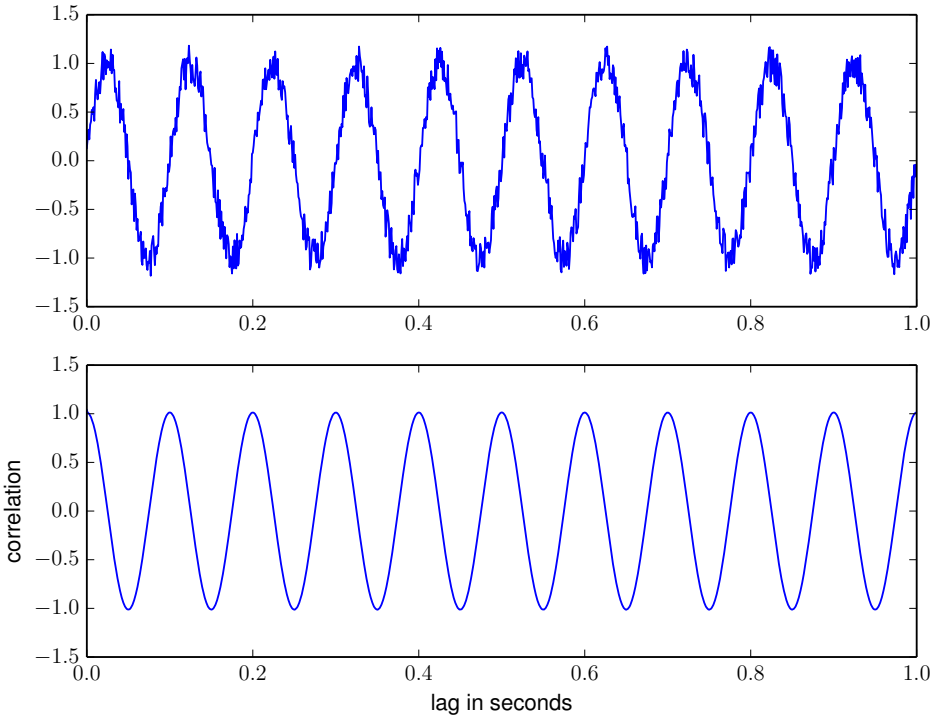

Figure 3.4: A noisy 10 Hz sine wave and its autocorrelation.

signal obscured by noise. Intuitively, if a signal is periodic with a frequency of 10 Hz, then when the signal is shifted with a lag that is a multiple of 1/10th of a second, the correlation should be higher than when it is shifted by a different lag. Figure 3.4 shows an example of how autocorrelation works for such a 10 Hz periodic signal. The top graph shows a 10 Hz sine wave with added noise over one second, while the bottom graph shows the cyclic autocorrelation of the sine wave. The graph shows how the autocorrelation of a periodic signal is itself a periodic signal. It also shows how the autocorrelation is highest when the time-lag is a multiple of 1/10th of a second, 0.1, 0.2, 0.3, and so on.

After autocorrelation is applied to the signal, Vilic et al. performed a FFT on the data, with zero-padding to achieve an FFT resolution of 0.1 Hz. The flickering targets on the monitor are considered classes, and each class has its own frequency. The score for each class, C_x, is calculated as the sum of power amplitudes, $|Y|$, within ± 0.1 Hz of the fundamental frequency, $H1$, and the second harmonic, $H2$:

$$C_x = \sum_{H1-0.1}^{H1+0.1} |Y| + \sum_{H2-0.1}^{H2+0.1} |Y| \tag{3.3}$$

All values of C_x are normalized with respect to one another with the dominating class having the value of 1. Selection happens if at least one of three conditions are met:

- The second greatest value in $SData < 0.35$.
- The second greatest value in $CData < 0.45$.
- The same class, C_x, is dominating in four consecutive iterations.

These thresholds were decided empirically. To test the system, users had to write four different sentences, and wrong input had to be corrected. The average ITR was 21.94 bits/min, and the average characters per minute (CPM) was 4.91, with the best case being 8.74.

SSVEP can be classified in three ranges: low (< 12 Hz), medium (12-30 Hz) and high (> 30 Hz). While higher frequencies evoke weaker SSVEPs, the SNR is similar for all three ranges, due to higher frequencies experiencing a decrease in spontaneous EEG (noise) [15]. Higher frequencies also cause less visual fatigue for users compared to lower frequencies, since the flickering effect is less visible. **Diez et al. [15]** made an SSVEP system using higher frequencies: 37, 38, 39 and 40 Hz. The experiments were carried out with flickering LEDs mounted on the four sides of a computer monitor, controlled by a field-programmable gate array (FPGA). A Grass MP15 amplifier system[3] was used for signal acquisition of the O1, O2 and Oz electrode placements with a sampling rate of 256Hz. Before online experiments were carried out, baseline data for each test user was collected; the users stared at a blank screen for 60 seconds. During the online experiment, the users were tasked with steering a ball through a maze on the computer monitor, with the four LEDs corresponding to the directions up, right, down and left. The signal processing is performed on a window with a length of two seconds (512 data points). A butterworth bandpass filter of order 6 is performed on the data with a cutoff of 32 Hz and 45 Hz. An estimation of the PSD is calculated by using an FFT on the two second window. The data is zero-padded to 1024 points, giving an FFT resolution of 0.25 Hz. The normalized power at each stimulation frequency is then calculated as the mean value of the power on each channel:

$$P(f_i) = \sum_{ch=1}^{M} \frac{\sum_{\Delta f} \hat{S}_{ch}(f_i \pm \Delta f)}{\sum_{\Delta f} \hat{B}L_{ch}(f_i \pm \Delta f)} \Bigg/ M \tag{3.4}$$

where $P(f_i)$ is the normalized power estimation for frequency f_i (i = 37, 38, 39 or 40 Hz). ch is the channel (O1, O2 or Oz) and $\sum_{\Delta f} \hat{S}_{ch}(f_i \pm \Delta f)$ is the sum of the band power for channel ch within $\pm\Delta f = \pm 0.25$ Hz of target frequency f_i. $\hat{B}L$ is the PSD calculated from the baseline EEG recorded prior to the experiment. M is the number of channels, three in this case. The two second window updates itself with new data every 0.25 s, and $P(f_i)$ is calculated for every update. A classification

[3] `https://www.grasstechnologies.com/`

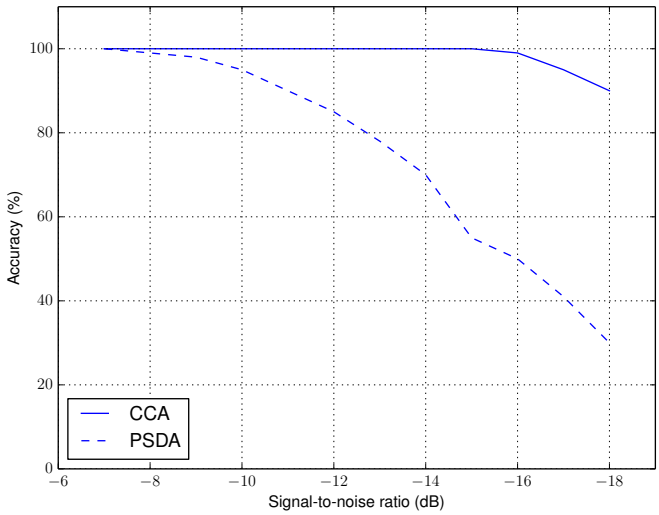

Figure 3.5: A plot of the accuracy using CCA and PSDA in an SSVEP-based BCI as a function of the SNR. Lin et al. [28] simulated noisy EEG signals by adding Gaussian white noise to sinusoidal waveforms. The plot shows how CCA handles decrease in SNR significantly better than PSDA. Adapted from Lin et al. [28].

is accepted if the maximum $P(f_i)$ is the same for a determined period of time H. H was adjusted on a user to user basis, between 1.5 s to 2.25 s. This system achieved ITRs ranging from 9.4 bits/min to 45 bits/min.

Canonical Correlation Analysis (CCA)

In the set of articles found through the SLR, many articles use a method called CCA in SSVEP-based BCIs. CCA can be used as an approach to analyze the frequency information in an EEG signal, and to find which frequency, among a set of frequencies, is the most prominent in the signal. CCA was first introduced for multi-channel SSVEP detection in 2006 by Lin et al. [28]. Lin et al. describe CCA as an array signal processing method that uses channel covariance information to extract frequency features in EEG. They performed a comparison between CCA and PSDA, and experienced the following: when the SNR decreased, the recognition accuracy of the PSDA approach decreased rapidly, whereas the CCA implementation retained a high accuracy. This is shown in Figure 3.5. They also experienced that the accuracy increased with the number of channels used for the CCA. The resistance to lower SNRs, and the ability to utilize information from many channels, are the main reasons why CCA is a popular technique in the BCI field.

Mathematically, CCA is a multivariate statistical method used when there are two sets of data which may have some underlying correlation [29]. The two sets of

data are put in multidimensional random variables X and Y. Consider the linear combination

$$X' = X^T W_x, Y' = Y^T W_y \tag{3.5}$$

X' and Y' are then projections onto W_x and W_y, and are called canonical variants. CCA finds the weight vectors W_x and W_y which maximize the correlation between X' and Y'. According to Bin et al. [3], this is done by solving the following problem:

$$\begin{aligned} \max_{W_x, W_y} \rho(X', Y') &= \frac{E[X'^T Y']}{\sqrt{E[X'^T X']E[Y'^T Y']}} \\ &= \frac{E[W_x^T X' Y'^T W_y]}{\sqrt{E[W_x^T X' X'^T W_y]E[W_x^T Y' Y'^T W_y]}} \end{aligned} \tag{3.6}$$

Solving this problem give the weight vectors W_x and W_y, but also several correlation coefficients. In the SSVEP-domain, it is common to use the largest correlation coefficient as a measure of correlation, as first suggested by Lin et al. [28].

When using CCA for frequency recognition in an SSVEP-based BCI, X is the multichannel EEG signal and Y is a reference signal. For each stimulus frequency, f_i, a reference signal Y_{f_i} is created. CCA will use Y_{f_i}, compare it to X and output a ρ_i which indicate the correlation between the two sets of variables (Figure 3.6). The stimulus frequency, f_s, which the user was visually fixated on, can be predicted by

$$f_s \approx f_c, C = \arg\max_i \rho_i \tag{3.7}$$

where C is the predicted target (command).

As stated above, the reference signal Y_{f_i} is a matrix created from a stimulus frequency f_i. The stimulus frequency, f_i, can be decomposed into the Fourier series of its harmonics [28]. The reference signal Y_{f_i} becomes the following

$$Y_{f_i} = \begin{pmatrix} \sin(2\pi f_i t) \\ \cos(2\pi f_i t) \\ \sin(4\pi f_i t) \\ \cos(4\pi f_i t) \\ \vdots \\ \sin(2\pi N_h f_i t) \\ \cos(2\pi N_h f_i t) \end{pmatrix}, t = \frac{1}{S}, \frac{2}{S}, \ldots, \frac{T}{S} \tag{3.8}$$

where N_h is the number of harmonics used, T is the number of sampling points and S is the sampling rate of the EEG acquisition device. $\frac{1}{S}$ is the length of each time step.

There were two articles found through the SLR that used CCA as their method to extract frequency information. **Bin et al. [3]** builds on the work of Lin et al.

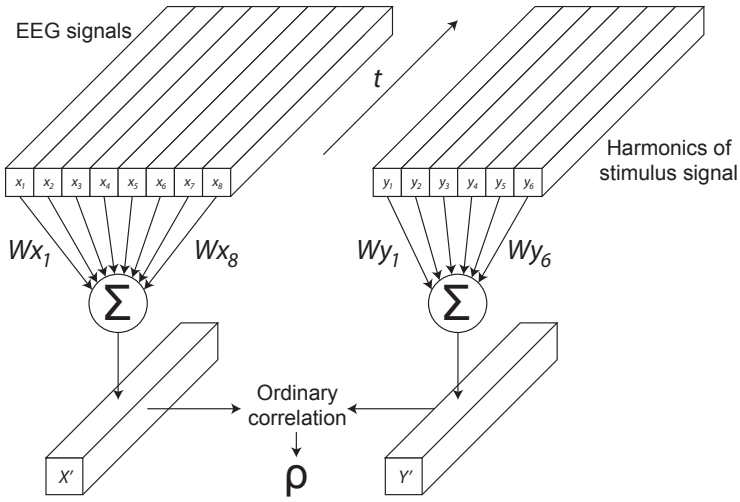

Figure 3.6: The usage of CCA for frequency recognition in an SSVEP system. $x_1, x_2, \ldots x_8$ are signals from 8 EEG channels, whereas $y_1, y_2, \ldots y_6$ are sine and cosine terms created from the harmonics of the stimulus frequency. CCA finds the weight vectors W_x and W_y which maximize the correlation between X' and Y'. ρ is the correlation between X' and Y'. Adapted from Lin et al. [28].

[28] and created a multichannel SSVEP system for online use. Bin et al. start by performing a general offline experiment to investigate the influence of the three important SSVEP parameters: channel location, window length, and the number of harmonics. For channel selection, Bin et al. used the method described in an article by the same authors [4]. This article describes how to apply CCA to create a topographic map of the scalp which can be used to guide the channel selection. CCA is run with the EEG data in X and with the correct stimulus frequency and its harmonics in Y. The correlation coefficients are not used. Instead, the weights W_x can be used to tell how important each of the channels are to get a resulting signal that correlates best with Y'. A channel that highly contributes to the correlation receives a corresponding weight, W_i, with high absolute value. Bin et al. [4] show how these channel weights can be mapped onto a figure of the scalp showing the channels according to the 10-20 system. The figure illustrates to what degree the different electrodes are contributing to the SSVEP signal.

Bin et al. reported strong activation of visual cortical areas in the occipital lobe with large positive and negative values. They conclude that the channels that lie in the areas near the occipital and parietal lobes (Figure 2.4), should be selected for EEG recordings in an SSVEP-based BCI. The sensors selected for the online system were O1, O2, Oz, PO7, PO8, POz, P3, P4 and Pz. Next, the number of harmonics and different time window lengths were tested on the offline data. The results showed that the number of harmonics had no significant influence on the

performance of the system. However, the accuracy of the system increased with the time window length. In the following online SSVEP system, only the first harmonic was used and the time window length was set to 2 seconds.

The online system used a 60 Hz LCD monitor as a stimulus device presenting six targets with frequencies 15 Hz, 12 Hz, 10 Hz, 8.6 Hz 7.5 Hz and 6.7 Hz respectively. The EEG signals were recorded using a BioSemi ActiveTwo system[4] with a sampling rate of 256 Hz. Each test subject was asked to input a string with 30 characters. The user had 0.3 seconds to shift his gaze between the 2 seconds recording interval. The system would output a predicted character every 2.3 seconds. The average of the correct count from 30 BCI commands was 28.6 giving an accuracy of 95.3%. The reported average ITR was 58 bits/min.

Cao et al. [6] also used CCA as their method to extract frequency information in their SSVEP-based BCI speller. On a 60 Hz monitor, the speller system enabled the user to choose among 42 characters (26 letters, 10 digits and 6 common used symbols). The characters were spread among three different pages with 16 targets in each page. Two of the 16 targets were used for turning the page. The frequencies of the 16 targets started from 8 Hz to 15.5 Hz with an interval of 0.5 Hz. The EEG signals were acquired with a g.USBamp amplifier from g.tec[5] with a sampling rate of 256 Hz. Six electrodes placed on POZ, P3, P4, OZ, O1, and O2 were used as input channels. The signals were filtered by a 0.5 Hz to 60 Hz band-pass filter.

The system operates with a gazing interval of 2 seconds, meaning that the user has to look at a target for 2 seconds for the system to predict an outcome. The following second is used for the user to shift his/her gaze to the next target. The window size is set to 1 second, and the correlation coefficients are calculated every 0.2 seconds. Throughout the gazing interval, the CCA algorithm runs 5 times producing 5 correlation coefficients and commands. At least 2 out of the 5 predicted commands, C, have to be the same. If this is not the case, no command is output from the system, and new EEG data has to be collected. If at least 2 commands are the same, the mean is calculated of the corresponding coefficients. If the mean is larger than the threshold 0.2, C is output as the prediction of the target. This system achieved an average accuracy of 98.78$\pm$ 1.62% and an average ITR of 61.64 $\pm$ 3.61 bits/min. The average spelling speed was 5.12 $\pm$ 0.22 seconds per character.

As opposed to the system described by Bin et al. [3], the system by Cao et al. had the possibility to not output any control command. The introduction of a threshold makes it possible to require the data to be of a given quality, before a command is given from the system. Cao et al. explains that a false positive of a system happens when the system accidentally performs an action that the subject did not intend. Having a significant false positive rate is undesirable in many practical BCI applications. Compared with false positives, it is better for the system

[4] `http://www.biosemi.com/`
[5] `http://www.gtec.at/`

to not output any control command when the user is gazing at targets, which is known as a false negative. A high threshold will reduce the false positive rate and lead to an increase in stability, but lower the sensibility of the system. Setting a threshold will be a tradeoff between sensitivity and stability. Cao et al. [6] reported an average false positive rate of 1.23% and a false negative rate of 3.06% in their system.

SSVEP with Adaptive Time-Window Length

In an SSVEP system, a crucial parameter is the length of the EEG segment that should be used to classify the stimulus frequency. The length of this interval will always be a tradeoff; a long interval gives higher accuracy, but increases response time, while a short interval results in a fast response time, but lower accuracy. Setting the correct length for this window can be difficult, especially when the system is to be used by multiple users. **da Cruz et al. [14]** proposed a speller system which could adapt the length of the time window simultaneously as the system was in use. The system allowed the user to input 48 characters as well as "Del" (delete) and "Undo". The four page system had 16 targets in each page, where 3 of the 16 buttons were reserved for turning page. The 16 targets, displayed on a 120 Hz monitor, were flickering with frequencies ranging from 8 Hz to 15.5 Hz, spaced out at intervals of 0.5 Hz. The "Del" button was selected by the user when correcting a misspelling as a result of a user error. The "Undo" button was selected by the user to tell the system that the last command was wrong, even though the user was gazing at the correct target. This gave feedback to the system about the online performance in real time. If many of the output commands are wrong, it could be an indication that the current window length is too short. The adaptive system can then increase the time-window length for the current user in real-time. For the contrary case, if the user does not send "Undo" commands for a longer period of time, the system will decrease the time-window length to lower the response time. In this way, the time-window length is adapted according to the online performance of the user.

In the proposed speller system, CCA was used as feature extraction in a similar manner to the approach described by Cao et al. [6]. The EEG signals were collected with a g.USBamp amplifier from g.tec[6] with a sampling rate of 256 Hz and filtered by a bandpass filter from 0.5-60 Hz. Six electrodes were used and placed at the POZ, PO3, PO4, OZ, O1, and O2 locations. The subjects were asked to input the same sequence of letters using both the fixed time-window length and the adaptive time-window length mechanism. The performance, averaged over all the subjects on both systems, are shown in Table 3.1. The adaptive time-window length mechanism gave lower detection time, and the improvement of ITR was 6.50% compared to the fixed time-window length system.

[6] `http://www.gtec.at/`

Time-Window Length Type	Accuracy (%)	Detection Time (s)	ITR (bits/min)
Fixed	100.00	4.09±0.24	58.36±3.28
Adaptive	99.00±1.15	3.85±0.13	62.09±2.31

Table 3.1: Average performance of adaptive time-window length mechanism vs fixed time-window length mechanism

Amplitude-modulated visual stimulation system

Chang et al. [10] investigated how amplitude-modulated (AM) stimulus could provoke SSVEP responses in a BCI system. The motivation behind creating an amplitude-modulated (AM)-SSVEP-system, is to combine the advantages of using high and low frequencies in an SSVEP-system. Chang et al. state that SSVEPs in the low-frequency band have a larger amplitude response than those in the medium- and high-frequency ranges, and that SSVEPs at 15 Hz have the largest amplitude. However, eliciting low frequency SSVEPs can be annoying and can cause epileptic seizures. High-frequency SSVEP systems are more comfortable for the user, causing low eye fatigue, but have significantly lower accuracy and ITR compared to low-frequency SSVEP-systems.

Amplitude modulation techniques are used in electronic communication, especially for radio carrier waves. The same principles are used when creating an AM stimulus for use in SSVEP. Chang et al. suggest that the brightness of the visual stimulus should vary as a double-sideband suppressed carrier (DSB)-AM sine wave, $S(t)$, which can be described by:

$$S(t) = c(t)m(t) = -\frac{1}{2}[\cos(2\pi(f_c + f_m)t) - \cos(2\pi(f_c - f_m t)] \tag{3.9}$$

As Equation 3.9 shows, $S(t)$ is generated from $c(t)$ and $m(t)$. Chang et al. refer to $c(t)$ as the carrier and $m(t)$ as the modulation signal. Both of these can be described by a sine wave:

$$c(t) = \sin(2\pi f_c t), m(t) = \sin(2\pi f_m t) \tag{3.10}$$

f_c is known as the carrier frequency and f_m the modulation frequency.

Chang et al. state that when the brightness of the stimulus varies as a DSB-AM sine wave and the stimulus is flickering with the carrier frequency, f_c, the maximum and minimum brightness of the stimulus will change sinusoidally at the modulation frequency f_m. By having the carrier frequency in the high-frequency band and the modulating frequency in the low-frequency band, the stimulus presents both high and low frequency information to the user.

To produce the stimuli, Chang et al. used two LED arrays (SMD 5050-3, Korea) with a diffusion film. The intensity variation of the six LEDs followed a similar

	Average accuracy (%)	**Average ITR (bits/min)**
AM-SSVEP	91.2	30.4
High-frequency-SSVEP	88.1	29.0
Low-frequency-SSVEP	86.7	29.1

Table 3.2: The performance of three types of stimuli for the Online 1 experiment.

	Average accuracy (%)	**Average ITR (bits/min)**
AM-SSVEP	97.02	39.41
High-frequency-SSVEP	94.95	38.17
Low-frequency-SSVEP	97.65	43.51

Table 3.3: The performance of three types of stimuli for the Online 2 experiment.

curve as the ideal AM stimulus described by $S(t)$. They used an ATmega128 from Atmel to digitize the $S(t)$ signal, and then an LTC1657CN digital-to-analog converter from Texas Instrument to convert the stimulus back into an analog signal to operate the LEDs. Various frequencies higher than 40 Hz were used as carrier frequencies. Low frequencies near the alpha-band were used as modulation frequencies. The EEG data was acquired with a g.USBamp amplifier from g.tec with a sampling rate of 512 Hz. The 15 electrodes that were used were located at these positions: O1, Oz, O2, PO3, POz, PO4, P1, Pz, P2, P3, P4, P5, P6, PO7, and PO8. During the measurement, a high-pass filter at 2 Hz, a low-pass filter at 100 Hz, and a notch filter at 60 Hz were applied to every channel.

CCA was used to extract frequency features from the produced EEG signals. An AM stimulus can be thought of as the sum of two sine waves of frequency $(f_c + f_m)$ and $(f_c - f_m)$. Harmonics of the frequencies $(f_c + f_m)$ and $(f_c - f_m)$ were used in generation of the reference signal, Y, for use in CCA.

In the online system, a window length of 4 seconds was used, shifting every 0.5 seconds to give a new segment of data. CCA was ran every 0.5 seconds, and if four consecutive predictions from the CCA were the same, a prediction was output from the system. Two types of online experiments were performed under different light conditions and different sequences of commands. Three types of stimuli were given: AM-, high-frequency- and low-frequency stimuli. The results for the Online 1 and Online 2 experiments are shown in Table 3.2 and Table 3.3 respectively. In the Online 1 experiment, AM-SSVEP performs the best with the highest ITR. In the Online 2 experiment, AM-SSVEP has higher ITR than high-frequency SSVEP, but lower ITR compared to low-frequency SSVEP.

A user evaluation was performed after the online experiments. The results from this evaluation showed that the users considered the low-frequency stimuli to give much higher eye fatigue and sense of flickering than high-frequency and AM-stimuli. Furthermore, the users regarded AM and high-frequency stimuli to be more suitable

for everyday use compared to low-frequency stimuli.

Minimum energy combination (MEC)

Volosyak [43] presented a method for detecting SSVEPs that uses MEC. First, consider an SSVEP response for a specific EEG channel (electrode) i: the voltage between the ith channel and the reference electrode at time t can be described as

$$y_i(t) = \sum_{k=1}^{N_h} (a_{i,k} \sin 2\pi kft + b_{i,k} \cos 2\pi kft) + E_{i,t}, \tag{3.11}$$

where N_h is the number of considered harmonics, f is the flicker frequency in Hz and $E_{i,t}$ is the part of the signal that cannot be attributed to the SSVEP response, i.e., noise. A time segment T_s with N_t samples for the ith signal can then be represented as a vector

$$y_i = Xg_i + E_i, \tag{3.12}$$

where $y_i = [y_i(1), ..., y_i(N_t)]^T$ contains the EEG signal over the time segment. X is the SSVEP information matrix that contains the sine and cosine components of the different harmonics used. g_i contains the corresponding amplitudes $a_{i,k}$ and $b_{i,k}$. For N_y electrodes, equation 3.12 becomes

$$Y = XG + E, \tag{3.13}$$

where $Y = [y_1, ..., y_{N_y}]$ contains the EEG signal for all channels (electrodes) over a time segment. Before applying MEC, remember that N_y different channels can be combined into one signal by using a spatial filter,

$$s = \sum_{i=1}^{N_y} w_i y_i = Yw, \tag{3.14}$$

where w is a set of weights $[w_1, ..., w_{N_y}]$, one for each channel. The purpose of weighting the different channels is to enhance the SNR. By applying different weights to the channels, several signals can be created from the same set of samples. For N_s signals, equation 3.14 can be generalized as

$$S = YW, \tag{3.15}$$

where $S = [s_1, ..., s_{N_s}]$ with the corresponding weight matrix $W = [w_1, ..., w_{N_s}]$. The first step of the MEC method is to remove the SSVEP activity from the recorded signal by using orthogonal projection:

$$\tilde{Y} = Y - X(X^T X)^{-1} X^T Y. \tag{3.16}$$

$\tilde{Y}$ contains mostly noise, artifacts and background brain activity. The second step of the MEC method is to find the weights, $\hat{w}$, for which $\tilde{Y}$ is minimized by optimizing the equation

$$\min_{\hat{w}} ||\tilde{Y}\hat{w}||^2 = \min_{\hat{w}} \hat{w}^T \tilde{Y}^T \tilde{Y} \hat{w}. \tag{3.17}$$

This will minimize the noise component of the signal in equation 3.14. Volosyak has in previous work shown that the lower bound of the right-hand side of the equation is given by the minimal eigenvalue λ_1 of the matrix $\tilde{Y}^T\tilde{Y}$ [17]. The corresponding eigenvector v_1 gives the weight vector for one signal and is the solution to the equation. Choosing more than one signal is based on the eigenvalues in ascending order with their respective eigenvectors, giving the weight matrix

$$W = \left[\frac{v_1}{\sqrt{\lambda_1}} ... \frac{v_{N_s}}{\sqrt{\lambda_{N_s}}}\right]. \tag{3.18}$$

Equation 3.19 shows how to find the total number of channels used, N_s.

$$\frac{\sum_{i=1}^{N_s} \lambda_i}{\sum_{j=1}^{N_y} \lambda_j} > 0.1. \tag{3.19}$$

Selecting enough channels that satisfies equation 3.19 will discard up to 90% of the noise in the signal [17].

After the signals have been filtered by the MEC method, Volosyak estimates the power of a frequency and its N_h harmonics by

$$\hat{P} = \frac{1}{N_s N_h} \sum_{l=1}^{N_s} \sum_{k=1}^{N_h} ||X_k^T s_l||^2. \tag{3.20}$$

In the experiments conducted in the article, $N_h = 2$. For N_f frequencies the power estimations are normalized into probabilities where

$$p_i = \frac{\hat{P}_i}{\sum_{j=1}^{j=N_f} \hat{P}_j} \qquad \text{with} \quad \sum_{i=1}^{i=N_f} p_i = 1, \tag{3.21}$$

where $\hat{P}_i$ is the power estimation of the ith frequency. A Softmax function is used to enhance the distance between the probabilities, defined by the function

$$p_i' = \frac{e^{\alpha p_i}}{\sum_{j=1}^{j=N_f} e^{\alpha p_j}} \qquad \text{with} \quad \sum_{i=1}^{i=N_f} p_i' = 1. \tag{3.22}$$

Volosyak [43] set $\alpha = 0.25$ based on empirical data when $N_f = 9$. The author further claims that higher α values reduces the time needed for command classification, but values higher than 0.3 produces many false positives and should be avoided.

In the online experiments, a speller was used to evaluate the performance with frequencies 6.67Hz, 7.5Hz, 8.57Hz, 10.0Hz and 12.0Hz used as target frequencies. These targets were presented on an LCD monitor with a 120 Hz refresh rate. Furthermore, four additional frequencies were considered to improve the robustness of

the classification. These frequencies were the means of two neighboring target frequencies: 7.08Hz, 8.03Hz, 9.29Hz and 11.0Hz. Classifier output O was accepted for frequency i if (1) ith frequency has the highest probability p'_i, (2) p'_i exceeds a predefined threshold β_i and (3) the detected frequency is one of the target frequencies. These three criteria can be summarized as

$$O = \begin{cases} argmax_i(p'_i) \\ p'_i \geq \beta_i \\ i \leq 5 \end{cases} . \tag{3.23}$$

β_i varies between 0.5 and 0.3 because visual stimulation with lower frequencies elicit a higher SSVEP than higher frequencies. The thresholds were decided empirically by analyzing offline data collected prior to this study. For the online experiment, EEG data was collected with the use of a g.USBamp from g.tec with a sampling rate of 128 Hz. The signals were filtered with an analog bandpass filter between 2 and 30 Hz and a 50 Hz notch filter. Pz, PO3, PO4, O1, Oz, O2, O9 and O10 were used for sensor locations in the 10-20 international system. The number of signals used by the MEC method is recalculated every 13 samples, or 101.5625 ms. 13 samples is considered one block. Classification is performed every 8, 20 and 40 blocks of EEG data, alternatively every 812.5 ms, 2031.25 ms and 4062.5 ms. If no classifier output is accepted after 8 blocks, the next calculations will be performed after 20 blocks, and then 40 blocks. The system achieved a mean ITR of 61.7 bits/min with a mean accuracy of 96.79%.

The MEC method has previously been used by **Cecotti [7]** in a speller system that achieved a mean ITR of 37.62 bits/min with an accuracy of 92.25%. It was also used by **Volosyak et al. [45]**, where a BCI speller system was tested on random subjects in real world conditions from volunteers at the international rehabilitation fair, RehaCare2008. They achieved a mean ITR of 22.6 bits/min and a mean accuracy of 92%.

3.1.2 c-VEP methods

This section gives an overview of the c-VEP methods discovered through the SLR. There were only two articles found that used c-VEP, most likely due to the fact that SSVEP methods are more popular, requiring no initial user training and having low implementation complexity. c-VEP, on the other hand, requires the user to look at a specific target in a training phase, requires the user gaze to be synchronized with the system and has higher implementation complexity. Advantages of a c-VEP system is high ITR and the possibility to place a larger number of targets within a given area without decreasing the performance of the system.

Bin et al. [2] describe how a simple c-VEP system can be built. A binary sequence can be used to represent a stimulus, where '1' and '0' represent 'light' and 'dark' respectively. Instead of having smaller repeated patterns of 0s and 1s, as is the case when representing a stimulus flickering with a frequency, the stimuli in a c-VEP-based BCI follow pseudorandom sequences. Bin et al. [2] modulate each target by a

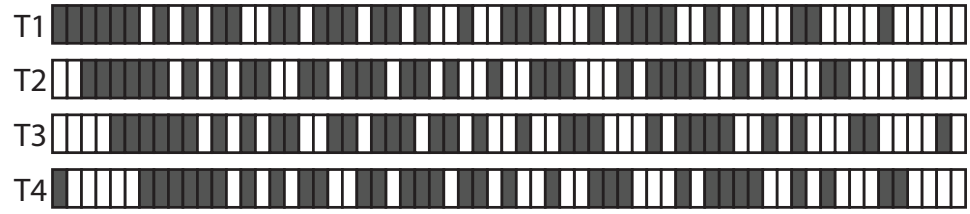

Figure 3.7: A 63-bit binary m-sequence. T1 is the original sequence, T2 is T1 circular shifted by two bits (frames), T3 is T2 circular shifted by two bits, and T4 is T3 circular shifted by two bits.

63-bit binary m-sequence. An m-sequence is a widely used pseudorandom sequence and its use is justified by Bin et al. [1]. The same 63-bit binary m-sequence was used for all targets, but the sequence was shifted two frames (bits), creating a two-frame time lag between two consecutive targets. An example is shown in Figure 3.7. Bin et al. [2] chose one target, T20, as the reference target. The reference template was obtained by averaging EEG data from the Oz channel in multiple stimulus cycles. In a stimulus cycle a stimulus will flicker according to its binary sequence, and repeat the same flickering in the next stimulus cycles. The length of a stimulus cycle will in general be the size of the bit string divided by the frequency of the monitor used. In the current article, a 60 Hz CRT monitor was used resulting in the length of the stimulus cycle to be 63 / 60 Hz = 1.05 s. This is also the length of the each template. The time lag between two consecutive targets is calculated as τ = 2 / 60 Hz = 0.033 s. Having the EEG data considered as the reference template, the templates for other targets were generated by circularly shifting the data by 0.033 seconds for each target.

Target identification was done by matching the incoming EEG data with each of the created templates. The best match gave the predicted target. Bin et al. described the target identification process in five steps:

1. In the training stage, the user is required to fixate on the reference target, T_r. EEG data within N stimulus cycles are collected as $x_n(t), n = 1, 2, \ldots, N$.

2. A reference template $M_r(t)$ is obtained by averaging over N cycles:

$$M_r(t) = \frac{1}{N} \sum_{n=1}^{N} x_n(t) \tag{3.24}$$

3. The templates of all targets are obtained by shifting the reference template:

$$M_k(t) = M_r(t - (\tau_k - \tau_r)), k = 0, 1, 2 \ldots, 31 \tag{3.25}$$

where $\tau_k - \tau_r$ indicates the time lag between target k and the reference target T_r.

4. For a segment of EEG data $x(t)$, the correlation coefficient p_k between $x(t)$ and the template $M_k(t)$ is calculated as

$$p_k = \frac{\langle M_k(t), x(t) \rangle}{\sqrt{\langle M_k(t), M_k(t) \rangle \langle x(t), x(t) \rangle}} \tag{3.26}$$

   where $\langle \mathbf{x}, \mathbf{y} \rangle$ indicates the product of $\mathbf{x}$ and $\mathbf{y}$.

5. The fixation target is identified by selecting the target that maximizes the correlation coefficient.

In this process the user gaze needed to be synchronized with the system. The user needed to look at a target from the start of the stimulus cycle and to the end. In the next stimulus cycle, the target identification was performed and the feedback was presented to the user. In this cycle the user could change his or her gaze to another target, and the data in this cycle was discarded. In the next cycle data was recorded, as the user was assumed to be gazing at a target.

To improve the c-VEP system, Bin et al. introduced the usage of multiple channels. CCA was used to generate spatial filter weights to be used for online data processing. In the training phase, the user gazed on the reference target, and multichannel EEG data within k stimulus cycles were collected and concatenated to a matrix $\mathbf{X}$ with dimensions $n \times (k * m)$. n was the number of channels and m the number of samples during a cycle. The average of the multichannel data from the k cycles was put in a matrix $\mathbf{R}$. For use in CCA a matrix $\mathbf{S}$, with the same dimension as X, was created by replicating $\mathbf{R}$ k times:

$$S = [R \quad R \cdots R] \tag{3.27}$$

The CCA method described by was set up with X and S. The formula 3.6 from Section 3.1.1 was applied to find the weights W_x for use as a spatial filter. The channels used by Bin et al. was O1, Oz, O2, P3, Pz, P4, PO7, POz, and PO8.

In the training stage, the users were gazing at the reference target for 200 stimulus cycles. The data was used to calculate the spatial filter weights and the reference template. Two systems were tested, one using 16 targets and the other 32 targets. When testing the two systems, each user was asked to input a sequence of 64 characters. Each selection required two stimulus cycles giving a response time of 2.1 seconds. The EEG signals were collected using a Synamps2 EEG system from NeuroScan Inc. Table 3.4 summarizes the results obtained when testing the two c-VEP systems.

Online adaptation of a c-VEP Brain-Computer Interface

Spüler et al. [38] created a c-VEP speller that adapts during use. The article considers two main approaches to online adaptation: unsupervised learning, and the use of error-related potentials (ErrPs). ErrPs are ERPs that can be detected

	Training accuracy (%)	**Online accuracy (%)**	**ITR (bits/min)**
16 target system	0.97±0.02	0.92±0.03	96±6.3
32 target system	0.93±0.04	0.85±0.05	108±12.0

Table 3.4: The performance of the two c-VEP BCIs. The presented results are the average from multiple users.

shortly after the user recognizes an error. When the system detects an ErrP, it knows that the letter/command that was output was wrong. Spüler et al. found that the ErrP for their BCI system has two main components: a small negative peak at around 310 ms after an error occurred and a larger, positive peak at 420 ms. These components were most prominent when using sensor locations Fz and Cz.

The c-VEP BCI system uses three steps to calibrate itself: first, training data is collected. The user gazes at one predetermined target, T_r, k times for k trials. Since c-VEP BCI targets use the same flicker sequence shifted in the time-domain, only one target is required for the training data. The second step generates a spatial filter based on the training data by using CCA. Spüler et al. differs from Bin et al. [2] in that the spatial filter is generated by finding a single channel C_b where the c-VEP is most prominent. This is done by using leave-one-out cross validation: for each trial, generate a template by averaging over the remaining $k-1$ trials, and the template with the highest correlation with the tested trial is selected. The accuracy for a single channel is calculated as the percentage of the correctly selected templates. This action is performed for all channels, and the channel with the highest accuracy is selected as C_b. CCA is then used to find the linear transformation W_x, W_y that maximizes the correlation between X' and Y', as shown in equation 3.6. In this case, X is the concatenated EEG data of all k trials, and Y is the desired waveform of the average c-VEP. Y is generated by computing the average of all k trials for channel C_b into a vector R and repeating the resulting vector R until X and Y have the same amount of samples in the time-domain. CCA is applied on X and Y, and the weights W_x are used as a spatial filter. The channels in the raw EEG data are multiplied by W_x, resulting in spatially filtered EEG data.

The third and final step in the calibration is to use a one class support vector machine (OCSVM) to generate the template for T_r, the original target the user was instructed to gaze at for k trials. The OCSVM is trained with the spatially filtered EEG data, and results in a hyper-sphere with minimal radius that encloses a given percentage of the data. The center of the hyper-sphere is used as the template M_r, representing the evoked response from the target T_r. Spüler et al. state that the use of a OCSVM is similar to using a method for averaging that rejects outliers, except it is more robust. From one template, M_r, the other templates for

other targets can be generated by shifting the template:

$$M_x(t) = M_r(t - \tau_s(x - r)) \quad x = 0, 1, ..., m \tag{3.28}$$

where τ_s is the time lag between two consecutive targets, m is the total number of targets and $0 \leq r \leq m$ is the target the template was created from. When classifying new data, the euclidean distance is calculated between the spatially filtered EEG data and all the templates. The template with the lowest euclidean distance is selected as the classifier output. Spüler et al. used LIBSVM[7] when implementing OCSVM, with a linear kernel and parameter $v = 0.5$.

To calibrate the classifier before using the system, a co-adaptive calibration approach was used: the system starts with a randomly generated template as a classifier, and the classifier is adapted under supervision to calibrate the BCI. After each user trial, the classifier is updated before the next trial. The correct target is known to the classifier.

After the initial calibrations to the classifier, the correct target is unknown when the user can freely decide what to type. Spüler et al. implemented an unsupervised online classifier adaptation. For a new trial D_x, the classifier outputs a label L_x. L_x is assumed to be correct, and the pair (D_x, L_x) is added to the training data. The classifier is retrained after each new trial in parallel to the system operations. Retraining the classifier involves finding the channel with the most prominent c-VEP, creating a spatial filter with CCA, training the OCSVM on all training data, and generating new templates for all targets. A problem with the unsupervised adaptation method, however, is that misclassifications are assumed to be correct. ErrPs can be used to detect when a misclassification occurs. If no ErrP is detected after a classification, the EEG data can be used for retraining. If an ErrP is detected, however, the EEG data is discarded since the true class label is unknown and the estimated label is suspected of being wrong. To detect ErrPs, EEG data from Fz, Cz, Cpz, Pz and POz were used for classification with LIBSVM using an RBF-Kernel with default parameters $C = 1, \gamma = 0.0091$.

For the online experiments, a monitor with a 60 Hz refresh rate was used. A matrix of $4 \cdot 8 = 32$ targets was presented on the monitor. For the target flickers, a 63-bit binary m-sequence was used, where additional targets were shifted by 2 bits. The length of one stimulation sequence is then $63/60 = 1.05s$. The resulting time lag is $\tau_s = 2/60 = 0.033s$ for two consecutive targets. EEG data was recorded with an EEG amplifier from g.tec, g.USBamp, collecting data at a sampling rate of 600 Hz. 30 electrodes were positioned on the scalp, focused mainly in the area on and around the visual cortex. The EEG signal was analogously bandpass filtered between 0.5-60 Hz, with a 50 Hz notch filter applied. The users of the system first had to go through co-adaptive calibration, for a total of 64 trials. For the unsupervised adaptation session as well as the ErrP-based adaptation, users were given words to type in beforehand. The results of both of these methods are summarized

[7] `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`

Method	Accuracy	ITR (bpm)
Unsupervised adaptation	92.53%	135.62
ErrP-based adaptation	96.18%	143.95

Table 3.5: Results of the online experiments performed with unsupervised and ErrP-based adaptation.

in Table 3.5. Spüler et al. explain the difference in the ITR between the two methods as inconclusive, since the results were gathered during different sessions. They performed additional offline tests with both methods on the same data, simulating an online test. In these findings, they found that the average accuracy for unsupervised and ErrP-based adaptation was 96.05% and 96.18% respectively. Further testing was done with the method proposed by Bin et al. [2], which achieved an accuracy of 88.48%. With the OCSVM approach used instead of the correlation approach suggested by Bin et al., an accuracy of 91.99% was achieved. Finally, OCSVM combined with the method for selecting the channel C_b with the most prominent c-VEP for use in the spatial filter as described above achieved an accuracy of 95%. Spüler et al. also found that the most prominent channels for c-VEP were P4 and PO3.

3.2 Pr2: Implementing a VEP-based BCI with Emotiv EPOC

Only one article that implements a VEP-based BCI with Emotiv EPOC passed through the SLR. **Liu et al. [29]** tested Emotiv EPOC due to the impracticality of medical grade EEG acquisition devices. High costs combined with a lengthy preparation time for use make medical equipment unfeasible for commercial use. Liu et al. conducted two different experiments: one offline experiment where they compared the Emotiv EPOC with medical grade hardware from g.tec, and one online experiment with Emotiv EPOC. For the offline experiment, an LCD monitor with 60 Hz refresh rate was used. Electrode placements O1, O2, P8 and P7 with a sampling rate of 128 Hz were used for both the Emotiv EPOC and g.tec hardware. CCA was used for classification of the offline data. The average accuracy went down by 11.8% with Emotiv EPOC compared to g.tec. The average ITR decreased by 7.6 bits/min, or 21.3%. The authors also point out that the subject who had the lowest accuracy and ITR of all subjects, had poor performance on frequencies in the range 8-13Hz. When analyzing the frequency spectrum, they found that the subject had a naturally strong response occurring in the alpha band (7.5-12.5Hz). They conclude that subjects with strong alpha waves can have worse performance when frequencies in the alpha band overlap with flickering frequencies in the SSVEP system.

For the online experiment, Emotiv EPOC was connected to MATLAB through software called BCI2000, which is a general-purpose system for BCI research cre-

Subject	Accuracy (%)	ITR	Average detection duration (s)
S1	100±0	22.57±0	4.78±2.62
S2	94.44±7.84	20.44±0.61	5.56±1.86
S3	100±0	23.98±0	5.27±2.42
S4	88.89±6.54	16.88±0.88	5.38±1.68
Mean	95.83±3.59	20.97±0.37	5.25±2.14

Table 3.6: Results of online tests with Emotiv EPOC

ated by Schalk Lab[8]. Every second, BCI2000 sends 128 samples to MATLAB for processing in form of a matrix. The samples are split into four, with a window length of 3 seconds updating every 0.25 seconds. CCA is performed on the window, and classification occurs if four predicted commands, C, in a row are the same. This means the minimum time before a classification can occur is four seconds. The results of the online experiment are shown in Table 3.6. The article also concludes that using existing software to connect Emotiv EPOC to MATLAB, in this case BCI2000, is troublesome and inconvenient, and it might be better to instead build a stable and flexible connection between Emotiv EPOC and MATLAB.

3.3 Pr3: Implementing a stimulus system for VEP-based BCIs

Visual stimulators in VEP-based BCIs are typically presented to users on a computer monitor or through LEDs. On a computer monitor, commands are represented by flickering targets with its own unique stimulus sequence. These targets are commonly presented as square boxes that alternate between two colors to create the flicker. An example of how a phone dialing paradigm could look on a computer monitor is shown in Figure 3.8. In an SSVEP BCI, each target flickers with a unique frequency and a flicker consists of an on/off cycle; a 10 Hz flicker consists of 10 on/off cycles per second. When representing these targets on a computer monitor, the range of available frequencies is limited by the computer monitor refresh rate. As an example, consider a monitor with a 60 Hz refresh rate, meaning it can draw a new frame 60 times per second. A 10 Hz flicker cycle would consist of six frames total, alternating every three frames between on and off. This can be calculated by

$$A = \frac{R}{f * 2} \tag{3.29}$$

where A is the alternating rate in frames, R is the monitor refresh rate and f is the frequency. A 10 Hz flicker on a 60 Hz monitor is illustrated in Figure 3.9. Now consider an 11 Hz flicker cycle. Representing an 11 Hz flicker cycle on a 60 Hz computer monitor is mathematically unfeasible since it would require the computer

[8] `http://www.schalklab.org/research/bci2000`

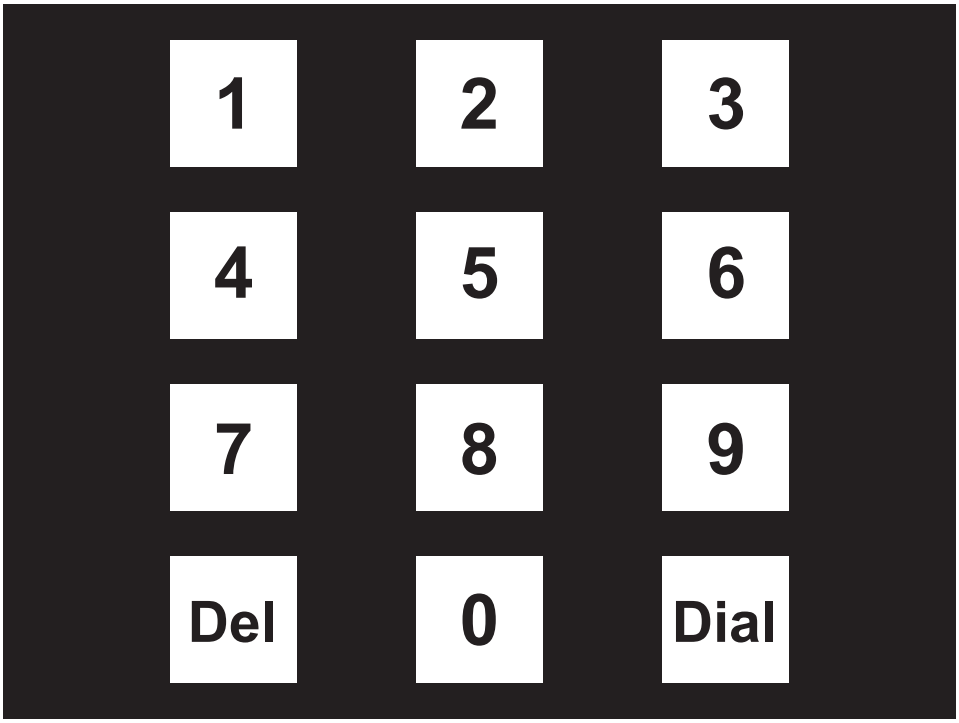

Figure 3.8: Example of how a phone dialing stimulator can look on a computer monitor. *Del* deletes the previous entry, and *Dial* dials the number entered so far.

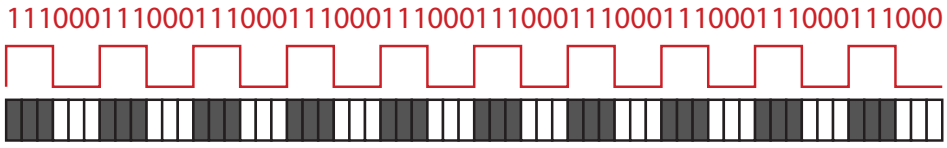

Figure 3.9: A 10 Hz flicker for a monitor with a 60 Hz refresh rate, shown as a bit string, a signal, and as frames. Dark frames means the stimulus is on, while light frames means the stimulus is off.

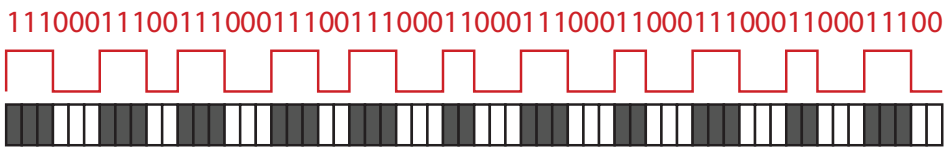

Figure 3.10: An 11 Hz flicker created by Equation (3.30) for a monitor with a 60 Hz refresh rate. It is shown as a bit string, a signal, and as frames. Dark frames means the stimulus is on, while light frames means the stimulus is off.

monitor to alternate between on and off every 2.73 frames according to Equation (3.29). This limits the possible frequencies on a 60 Hz monitor to 1 Hz, 2 Hz, 3 Hz, 4 Hz, 5 Hz, 6 Hz, 6.67 Hz, 7.5 Hz, 8.57 Hz, 10 Hz, 12 Hz, 15 Hz, 20 Hz and 30 Hz. The lower frequencies (< 5 Hz) are not practical to use in an SSVEP paradigm due to noise and high response time. Some combinations of frequencies should also be avoided in an SSVEP-based BCI since the fundamental frequency can elicit even stronger responses in the second and even third harmonic. For example, since 15 Hz is the second harmonic of 7.5 Hz it would be inadvisable to use both of these frequencies for flickers in an SSVEP system. **Wang et al. [46]** claim that visual stimulator design is currently the limiting factor for applications in SSVEP BCIs, and experimented with how to represent more frequencies on a computer monitor. The idea was to approximate frequencies by using a varying number of frames in each cycle, to represent frequencies that otherwise could not be represented. For an 11 Hz flicker, the cycles would vary between five and six frames, corresponding to 12 Hz and 10 Hz respectively. A flicker at frequency f at frame i can be calculated by:

$$flicker(f, i) = square(2\pi \cdot f \cdot \frac{i}{RefreshRate}) \tag{3.30}$$

where $square(2\pi ft)$ generates a square wave with frequency f and i is the frame index. For an 11 Hz flicker on a 60 Hz monitor, the resulting flickering sequence can be described by [3 3 3 2 3 3 3 2 3 3 2 3 3 3 2 3 3 3 2 3 3 2]. Each number corresponds to the number of frames, where pairs of two comprises one flicker cycle. The first cycle would be three on three off, while the next cycle would be three on two off, and so on. Figure 3.10 shows how the stimulus would look. To verify the method, Wang et al. performed offline analysis on EEG data collected from a user who gazed at an 11 Hz flicker on a 60 Hz monitor. Through PSDA they were able to verify that the flicker elicited a VEP at 11 Hz and 22 Hz in the frequency spectrum. The setup was then tested in an online setting as well, where users were asked to enter their phone numbers by gazing at targets on the monitor. EEG data was collected at a sampling rate of 256 Hz, and CCA was used for classification. The window length was set to 1.5 seconds, and updated itself every 0.2 seconds. A classification was accepted if the highest correlation coefficients were the same for three periods in a row. The experiment achieved an average ITR of 75.4 bits/min, with an average accuracy of 97.2 %.

	White	Gray	Red	Green	Blue
Accuracy (%)	96.25± 4.52	92.92± 7.88	88.75± 10.37	87.50± 11.12	85.00± 13.77
ITR (bits/min)	36.61± 3.88	34.20± 6.10	31.26± 7.31	30.44± 7.59	29.03± 8.92

Table 3.7: Results of different colors used for stimulation in SSVEP.

Cao et al. [5] conducted an experiment on how colors affect the performance of an SSVEP system. The theoretical background for the experiment has basis in how the human eye perceives color and light. The human eye contains two types of photoreceptors, namely rods and cones. The rods are extremely sensitive to light, but cannot differentiate between colors. Cones are less sensitive to light, and split into three different types: those that perceive red, those that perceive green and those that perceive blue. Experiments were ran for five different colors: white (255, 255, 255), gray (128, 128, 128), red (255, 0, 0), green (0, 255, 0) and blue (0, 0, 255). The numbers in the parentheses correspond to the RGB values for that color. CCA was used as the classification method. The article found that the correlation coefficients were higher for white and gray colors, followed by red, green and finally blue. These findings were supported in the subsequent online experiment. The ITR and accuracy of the different colors are shown in Table 3.7. Cao et al. conclude that white and gray outperform the other colors because they can simultaneously elicit all three cone types (red, green and blue), whereas the other colors only elicit one type of cone. This results in a more intense SSVEP.

da Cruz et al. [13] experimented with patterned visual stimuli and its effect in an SSVEP system. The goal was to see if patterned stimuli could enhance the SNR and ITR. For each flickering object on a computer monitor, a blue cross was placed in a random position on top of it, up to a maximum of 12 crosses per flickering object. In the online experiment, the patterns were cleared when a command was detected or when the limit of 12 crosses was reached. The positions of the crosses were randomly generated every 0.5 sseconds, meaning 12 crosses were generated in six seconds. The x and y position of the cross was given by

$$[x, y] = [r \cdot \cos\theta, r \cdot \sin\theta] \tag{3.31}$$

where r and θ were randomly generated every 0.5 seconds for each flickering object. During the offline experiments, an average increase of 43.6% was seen in the SNR from using patterned visual stimuli. CCA was used as a classification method in the online experiment, where users were asked to type different words in an SSVEP speller. The users stated that the patterned visual stimuli was more comfortable to look at than non-patterned stimuli. In addition, an average ITR of 53.6 bits/min was achieved with patterned stimuli compared to 45.9 bits/min with non-patterned stimuli.

3.4 Discussion

The articles presented above are heavily favored towards SSVEP-based BCIs, for reasons explained in Section 3.1.2. There were no t-VEP articles found through the SLR. As discussed in Section 2.2.1, t-VEP-based BCIs have inherent limitations that restrict the effective throughput of the method, which could explain the lack of research for this approach in modern times. In this section, the key points from the related work that was presented in this chapter are discussed. The section concludes with a justification for the VEP techniques that are used in this thesis. Note that research question 4 (RQ4) (see Section A.1) of the SLR is answered throughout this section.

3.4.1 VEP techniques

PSDA was for a long time the most common technique for analyzing SSVEPs. One of the main advantages of this method is that it is simple to implement, and it is useful for seeing if there is an evoked response as a result of a periodic stimulus. PSDA provides a good basis of comparison for other methods. One of the disadvantages of PSDA is the low ITR when compared to other, more recent methods.

One of the most used new methods is CCA, which has been applied in multiple ways in the BCI field. As PSDA, CCA can be used for extraction of frequency information. CCA is, however, more resistant to noise, allows the use of multiple channels and has better ITR. In addition, the CCA-weights can be used both for channel selection and as a spatial filter. Channel selection and spatial filtering can be used as a preprocessing step for other techniques, as shown by the c-VEP systems. Another method that also outperformed PSDA, is MEC. MEC attempts to increase the SNR in a signal by reducing the noise component. Systems using MEC perform similarly to the ones using CCA, but has a higher implementation complexity.

Chang et al. [10] presented a different type of BCI system which made use of two frequencies, one low and one high, to produce stimuli. The goal was to have a system causing less fatigue, but still have about the same performance as a low frequency SSVEP. The obtained ITR was not significantly better than a regular high-frequency SSVEP, and the proposed AM-SSVEP system has a high implementation complexity.

Of all the VEP techniques mentioned in this chapter, c-VEP had the best reported ITR. Spüler et al. [38] obtained the highest reported ITR in a non-invasive BCI. A c-VEP system has higher implementation complexity than an SSVEP system and requires a training phase for using the system, but the gain of implementing such a system is high. One of the reasons for the high ITR is the ability to use more targets than in an SSVEP system.

da Cruz et al. [14] showed how the window length could be adjusted in real-time

while the SSVEP system is running. The window length was adjusted according to the performance of the current user. The article illustrates the importance of having a suitable window size, and the benefit of having an individual window size for each user. For a c-VEP system, it would be difficult to have unique window lengths as the window size follows the length of the bit string used.

3.4.2 Visual stimulator and Emotiv EPOC

The visual stimulator is tasked with providing accurate stimuli to create a strong VEP in the user. Wang et al. [46] showed that it is possible to represent frequencies on an LCD monitor that are not divisible by the monitor's refresh rate, allowing for more potential target frequencies in an SSVEP system. Cao et al. [5] concluded that using white color for stimulation produces a more intense SSVEP due to eliciting a response from all three color cones in the human eye (red, green and blue).

Emotiv EPOC performs significantly worse than a medical grade EEG acquisition device. However, Liu et al. [29] showed that it can be used together with CCA to classify SSVEPs. Liu et al. concluded that it is cumbersome and restrictive to connect Emotiv EPOC to MATLAB with current existing software, e.g. BCI2000. They also found that users with strong alpha waves can cause problems in an SSVEP-based BCI when the target frequencies are in the same range as the alpha band (8-12.5 Hz).

3.4.3 Choice of VEP techniques

In this thesis, four VEP methods are implemented and compared to one another based on the SLR conducted.

The first of these methods is the PSDA method presented by Vilic et al. [42]. One of the reasons for choosing a PSDA method is because SSVEPs can be analyzed by PSDA to confirm that Emotiv EPOC produces a periodic VEP given periodic stimulus. The second reason is that PSDA can be used as a baseline for comparison for the other methods, since it was the most used SSVEP method prior to the introduction CCA. The reason for choosing the method presented by Vilic et al. [42] over the one presented by Diez et al. [15] is because Diez et al.'s method uses high frequencies (37-40 Hz). The LCD monitor used in this thesis has a refresh rate of 60 Hz, so the maximum frequency that can be represented is 30 Hz.

The second method is the CCA approach described by Cao et al. [6]. This is a more recent implementation of CCA that had a higher ITR than the one presented by Bin et al. [3]. Emotiv EPOC has a lower SNR compared to medical grade equipment, and CCA has been shown to be a robust technique with a high tolerance for noise (see Section 3.1.1). In addition, Liu et al. [29] has already demonstrated that an SSVEP can be identified by a CCA method when using Emotiv EPOC. The CCA method described by Liu et al. [29] will also be implemented and tested. This method is slightly different then the one presented by Cao et al., and it will

Identifier	Original author	Feature extraction	VEP paradigm
M1	Liu et al. [29]	CCA	SSVEP
M2	Cao et al. [6]	CCA	SSVEP
M3	Vilic et al. [42]	PSDA	SSVEP
M4	Bin et al. [2]	Template matching	c-VEP

Table 3.8: The VEP methods chosen for implementation and testing.

be interesting to see how these two compare.

The fourth and final VEP method will be the c-VEP method by Bin et al. [2]. This algorithm gives lower performance than the one presented by Spüler et al., but is simpler to implement. It is expected that the simpler algorithm will tell whether or not c-VEP is suitable for use with Emotiv EPOC. The SLR did not give any indications that c-VEP has been used with Emotiv EPOC before. If this technique shows that a c-VEP response can be obtained, then the more advanced method by Spüler et al. will also be implemented.

Table 3.8 gives an overview of the methods chosen for implementation and testing.

Chapter 4

System Implementation

This chapter details the implementation of the visual stimulator used to produce stimuli for the users, and how to connect to Emotiv EPOC (Section 4.1). In Section 4.2, the practical difficulties with using Emotiv EPOC are discussed. The chapter concludes with a thorough explanation of how the different methods selected in Section 3.4.3 were implemented.

4.1 Python System

Python was the programming language that was used for creating a VEP-based BCI system. Python was chosen for several reasons:

Speed of development
: Python is a dynamically typed interpreted scripting language. It offers functionality for object-oriented programming and structured programming, and to a lesser extent functional- and aspect-oriented programming. Since it is an interpreted scripting language, it is easy to develop and test code rapidly.

Open source libraries
: Python offers a range of useful open source libraries. The two most important libraries for scientific purposes are NumPy[1] and SciPy[2]. These libraries offer efficient n-dimensional arrays, as well as an extensive selection of optimized and useful calculations, such as FFT and linear algebra functions.

One concern with using Python as a programming language was the issue of speed. If the calculations performed to process and classify EEG data are too demanding on the system, it can result in the loss of frames for the visual stimulator, which can result in poor VEPs. Figure 4.1 shows the class diagram for the completed system. The rest of the section will explain in detail the different parts of the system.

[1] `http://www.numpy.org/`
[2] `http://www.scipy.org/`

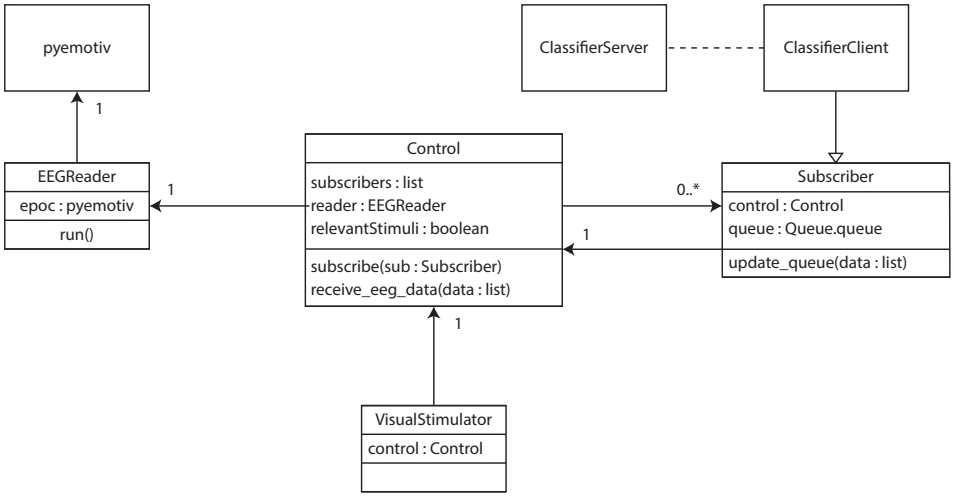

Figure 4.1: The complete class diagram of the implemented Python system. The stippled line between ClassifierClient and ClassifierServer denotes a network connection.

4.1.1 Connecting to Emotiv EPOC

The SDK that comes with Emotiv EPOC provides two dynamic-link libraries (DLLs) that can be used to communicate with the headset. These libraries allow access to the raw EEG data from the headset, but some pre-processing is performed prior to this. For each EEG sample, the value of the reference channel is subtracted from each channel, and digital notch filters are applied at 50 and 60 Hz (see Table 2.1). To use the DLLs, an open source Python library called pyemotiv[3] was used. Pyemotiv connects to the headset and sets the data buffer to contain the last five seconds of data. When pyemotiv requests data from the headset, the headset returns the samples that accumulated since the last time pyemotiv requested data. By empirical testing, it was confirmed that Emotiv EPOC will not return any data samples until it has accumulated a minimum of four samples. A sample in this context contains the multichannel EEG data for one timestamp.

The Python system is mostly object oriented, with some functionality offered by modules. The data acquisition is performed by a class called EEGReader. EEGReader uses the pyemotiv library to continuously request raw EEG data from Emotiv EPOC. When new data is received, EEGReader sends the data to a controller class called Control. When other objects want access to the EEG data, they have to subscribe to Control. Control keeps a list of subscribers, and when new data is received from EEGReader, Control will send the data to all subscribers. Since all classes run in separate threads, each subscriber must use the built-in Python

[3] `https://github.com/thearn/pyemotiv`

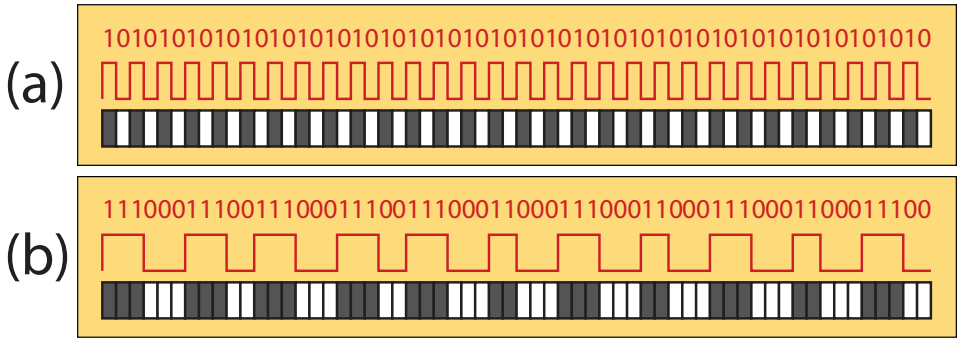

Figure 4.2: (a) shows a 30 Hz stimulus sequence, (b) shows an 11 Hz stimulus sequence.

Queue class. The Queue class implements a First in, First Out (FIFO), thread-safe producer/consumer queue which Control updates with new data. Figure 4.1 shows how this works.

4.1.2 Visual stimulator

An open source Python library called PsychoPy[4] was used for the presentation of visual stimuli on a computer monitor. PsychoPy allows the presentation of stimuli and collection of data for a wide range of neuroscience, psychology and psychophysics experiments, and uses OpenGL to draw the visual stimuli. A Window object, which is an OpenGL context, has a function called *flip*. This function swaps the front and back buffers. The front buffer is the buffer that is currently displayed on the monitor, while the back buffer is the buffer that is being drawn to. Calls to *Window.flip()* are synchronized to the monitor refresh rate, and the function call will block until the swapping of buffers has occurred. This ensures accurate timing information for individual frames as long as frames are not being dropped. When a frame is dropped, it means that the program is making consecutive calls to *Window.flip()* slower than the monitor refresh rate. PsychoPy provides the exact functionality for what is needed to produce accurate VEP stimuli. VEP stimuli is represented as a bit sequence of variable length, where each bit represents whether the target should be on (1) or off (0) for that particular frame. As an example, consider a target that flickers at 30 Hz. Figure 4.2(a) shows how a 30 Hz flicker on a monitor with a 60 Hz refresh rate is represented as a bit sequence. As discussed in Section 3.3, Wang et al. [46] presented a technique for displaying frequencies that are not divisible by the monitor refresh rate, by using Equation 3.30. The equation was implemented with SciPy's square function. Figure 4.2(b) shows how an 11 Hz target flicker can be represented on a computer monitor by using this method.

[4] `http://www.psychopy.org/`

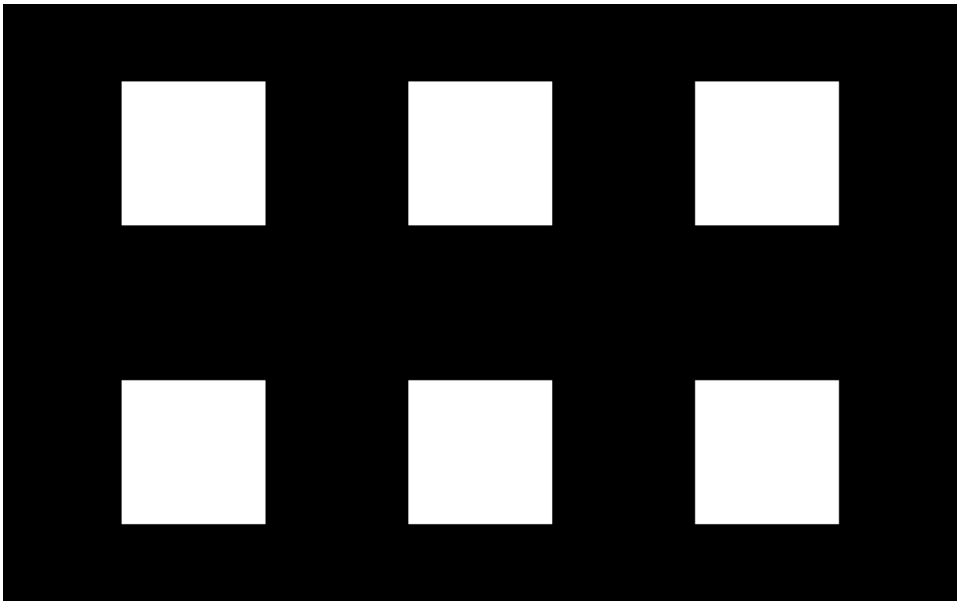

Figure 4.3: The visual stimulator with six white targets.

The visual stimulator consists of six targets (boxes) that each have their own individual flicker sequence. The boxes are spaced out with three boxes on the top row and three boxes on the bottom row, as shown in Figure 4.3. The boxes are spaced far from each other to avoid interference from the targets that the user is not gazing at.

Since several of the VEP techniques that were chosen in Section 3.4 require the classifiers to know when the visual stimulator is showing relevant stimuli, it is beneficial to have both the visual stimulator and the classifiers running in the same process. An OpenGL context (the visual stimulator) has to run in a process' main thread, so all other functionality, like subscribers, has to run in separate threads. Since the classifiers are only interested in the EEG data from when the visual stimulator is showing the stimuli, the visual stimulator informs the Control object of when it is showing relevant stimuli. The Control object in turn will only send data to its subscribers if the signal for relevant stimuli has been received.

4.1.3 Online analysis

After running initial experiments, it was found that performing CCA on EEG data in real-time required a large amount of central processing unit (CPU) power. This caused several frames to get dropped from the visual stimulator, due to a combination of intensive computations and Python being a slow language. When frames get

dropped, the stimuli have smaller chances of eliciting the desired VEPs in the user. Instead of performing all computations on one computer, a simple client/server pair was set up. The server receives EEG data over a TCP connection, and the packet header specifies what computations should be performed on the data. The result is then returned to the client. Since both the client and server are on the same network, the latency is less than 1 ms and the overhead related to making the EEG data serializable is considered insignificant in terms of performance.

4.2 Practical Considerations

The EEG values from the sensors of Emotiv EPOC have a direct current (DC) offset. The DC offset is the mean value of the waveform for each channel. According to gmac, a representative from Emotiv, there are two contributions to the DC offset[5]:

1. The first of these is a constant offset caused by Emotiv EPOC sending the EEG data as an unsigned integer. This offset puts the mid-range point of the data at 4096 units.

2. The second contributor comes from the body potential. The body potential drifts around based on what a person is doing. Emotiv EPOC attempts to stay centered on the body's average potential, but there is a latency for how quickly it can react. This manifests itself as a slow DC drift in the signal.

Since the DC offset drifts based on body potential, an infinite impulse response (IIR) filter is applied to the EEG data from Emotiv EPOC, similar to a running average. The purpose of the filter is to identify and remove the background noise such as the DC offset. The implementation is based off another forum post by gmac[6]. The first samples received are set as the current background noise. Pseudocode for the IIR filter is shown in Algorithm 1. IIR_TC is the memory of the IIR filter, and is set to 800 samples, or 6.25 seconds. Effectively it is the same as using a 0.16 Hz high-pass filter.

Another consideration with Emotiv EPOC is that the headset is not adjustable to accommodate different head shapes. Since the reference channel is subtracted from all other channels as discussed in Section 4.1.1, good sensor connectivity for the reference channel is important to achieve accurate sensor readings for all channels. When the reference sensor has poor connectivity, it is extremely sensitive to external noise. As a demonstration, Emotiv EPOC was used by both authors of this thesis, and it only had a good fit for one. EEG data was recorded while a person was walking behind them as a source of external noise. Figure 4.4 shows the results of the short experiment. Subject 1 had a good fit for the reference sensor, while Subject 2 did not. The graph clearly demonstrates the importance of a good fit for the reference sensor for accurate sensor readings. One option to ensure good

[5] `http://emotiv.com/ideas/forum/messages/forum15/topic1338/message11928/#message11928`

[6] `http://www.emotiv.com/forum/messages/forum15/topic984/message6466/#message6466`

Algorithm 1 IIR filter

$IIR\_TC \leftarrow 800$
while samples from headset **do**
  for all channels **do**
    if first sample **then**
      $background \leftarrow sample$
    else
      $background \leftarrow \frac{background \cdot (IIR\_TC - 1) + sample}{IIR\_TC}$
    end if
    $sample \leftarrow sample - background$
  end for
end while

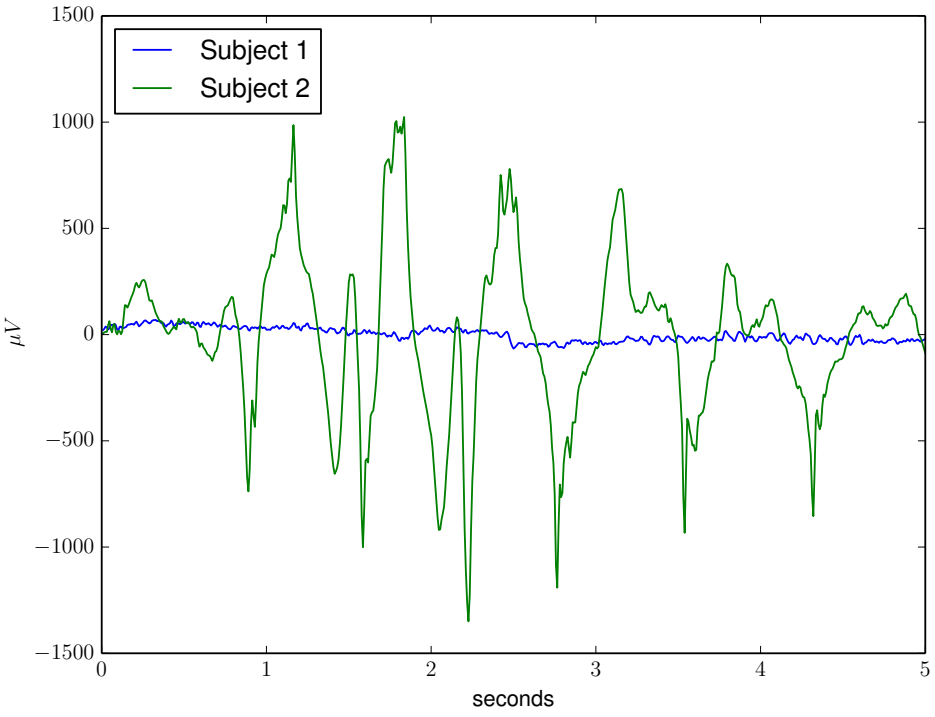

Figure 4.4: Example of how a bad fit for Emotiv EPOC causes external noise in the EEG signal. Subject 1 has a good fit while Subject 2 does not. The external noise was created by having a person walking behind the test subjects.

connectivity for the reference sensor is to adjust the headset by pulling it forward until the reference sensor has a good fit. By doing so, the remaining four sensors that cover the O1, O2, P7 and P8 positions are also pulled forward, causing poor coverage of the visual cortex. VEPs are mainly read from the sensors over the visual cortex, and adjusting the headset so the sensors are pulled away from the visual cortex will result in poor performance.

4.3 Implemented Algorithms

In Section 3.4.3, four methods were selected to be implemented and tested, all shown in Table 3.8. This section gives an overview over how the different methods were implemented, and how the methods were adapted to the hardware used in this thesis. In Section 2.2.3, ITR was introduced as a performance metric to evaluate BCI methods. Since the number of targets is one of the variables that determines the ITR, all methods use six targets to ensure that a fair comparison between them can be made.

4.3.1 M1 using CCA

Liu et al. [29] implemented a method that uses CCA for classification with Emotiv EPOC. An LCD monitor with a 60 Hz refresh rate was used for the experiments, with electrode placements O1, O2, P7 and P8, shown in Figure 4.5. The method uses a moving window length of three seconds, and it calculates a correlation coefficient every 0.25 seconds. If four coefficients in a row belong to the same target frequency, that target is selected. BCI2000 is used to send the EEG data (in blocks of 128 samples) from the headset to MATLAB, where the coefficients are calculated. That means the minimum time before a target can be classified is four seconds. Liu et al. found the connection between the Emotiv headset and MATLAB to be a troublesome and inconvenient solution.

Implementation

MATLAB has a function *canoncorr*[7] that implements CCA. The implementation is viewable through MATLAB as well as online[8]. The function takes as input two sets of data, X and Y, and returns five variables. It returns the weights for X and Y that maximize the correlation, the canonical correlation coefficient, as well as the weighted data for X and Y (the data sets multiplied by the weights). The function was translated from MATLAB into Python by using equivalent functions in NumPy and SciPy to do the calculations. Out of the five returned variables, only the correlation coefficient is needed to perform classification. The correctness of the implementation was verified by comparing the output to the output from MATLAB and ensuring that it was the same.

[7] `http://www.mathworks.se/help/stats/canoncorr.html`

[8] `http://affect.media.mit.edu/projectpages/affective_cognitive_decision_making/DM%20Experiment/stats/canoncorr.m`

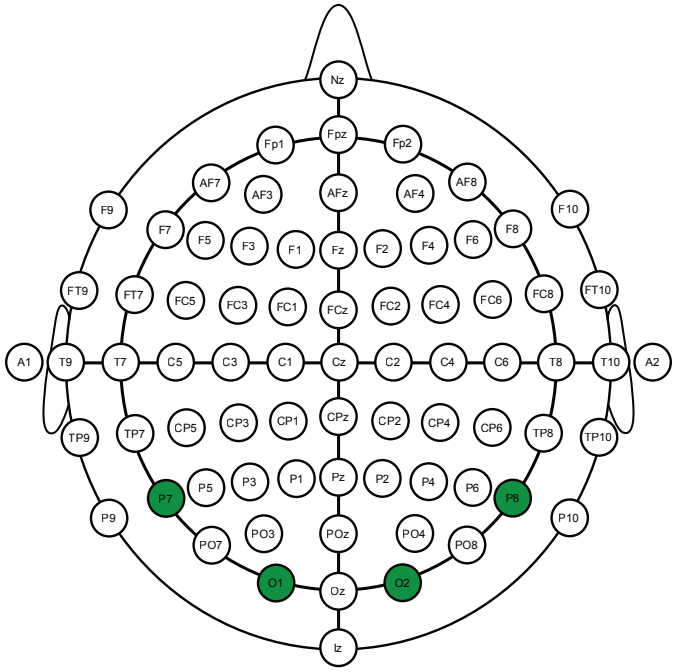

Figure 4.5: The EEG channels used for the M1 method are shown here in green. Adapted from Marius 't Hart - `http://www.beteredingen.nl`. Used under CC BY: `http://creativecommons.org/licenses/by-sa/3.0/nl/deed.en_GB`.

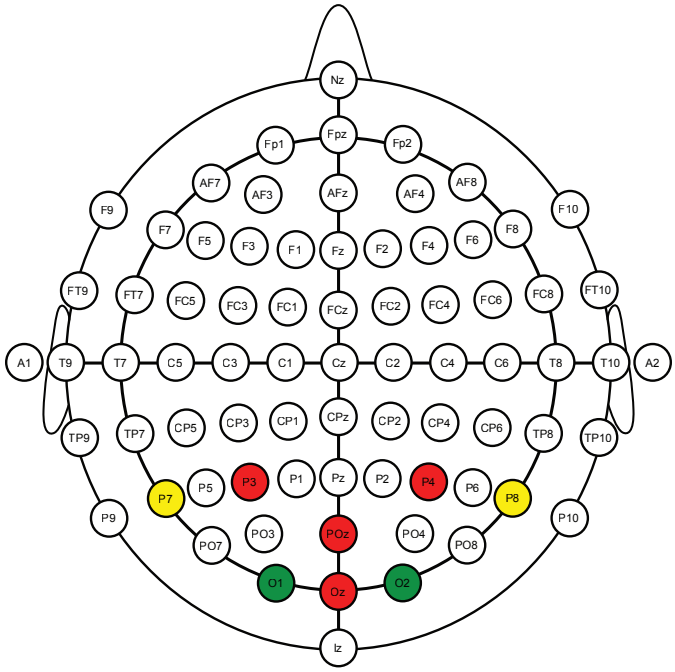

Figure 4.6: The EEG channels used for the original M2 method compared to the ones used by Emotiv EPOC. The shared channels are green, the ones only used by Emotiv EPOC are yellow, while the ones only used by the original method are red. Adapted from Marius 't Hart - `http://www.beteredingen.nl`. Used under CC BY: `http://creativecommons.org/licenses/by-sa/3.0/nl/deed.en_GB`.

The method is implemented in almost the exact same way as the original implementation. Since the data is available immediately in the Python program, that means the minimum time for classification to occur is 3.75 seconds when the window length is 3 seconds.

4.3.2 M2 using CCA

Cao et al. [6] presented an SSVEP method that uses CCA to identify the target frequency the user is gazing at. They tested it on a spelling system that used an LCD monitor with a 60 Hz refresh rate. 16 targets were presented on the monitor at a time and used frequencies from 8-15.5 Hz at intervals of 0.5 Hz. EEG samples were acquired with a 256 Hz sampling rate, and six EEG channels were used: POz, P3, P4, Oz, O1 and O2. The difference between the used channels between this method and Emotiv EPOC are shown in Figure 4.6. The signals were filtered with a bandpass filter from 0.5-60 Hz.

The method uses a moving window length of 1 second, and operates with a gazing interval length of 2 seconds. The method then calculates the CCA coefficients every 0.2 seconds for all targets after the first second at 1.2, 1.4, 1.6, 1.8 and 2.0 seconds, and the maximum coefficient out of all targets is saved for a total of five coefficients in two seconds. If two of the five coefficients or more are for the same target frequency, the average value of the coefficients is calculated. If the average value is larger than 0.2, that target frequency is selected.

Implementation

A difference between the implementation by Cao et al. [6] and the one used for this thesis, is that instead of calculating five correlation coefficients at 1.2, 1.4, 1.6, 1.8 and 2.0 seconds, they are calculated at 1.0, 1.25, 1.5, 1.75 and 2.0 seconds. The reason for this is the sampling rate of 128 Hz by Emotiv EPOC, which is not divisible by five but is divisible by four. It should also be noted that all the experiments conducted in this thesis use only six targets, whereas Cao et al. [6]'s method uses 16 targets. A result of this is that each target frequency has an increased chance of being the highest correlation coefficient when there is a lack of frequency information in the EEG data. Therefore, if a target frequency has the highest correlation coefficient more than twice out of the five, only the two highest coefficient values are used to calculate the average. This is done to avoid the randomly selected correlation coefficients from interfering with the average when there is little to no frequency information in the signal. The same threshold of 0.2 in the original method is used.

4.3.3 M3 using PSDA

Vilic et al. [42] presented a thresholding method that uses PSDA to recognize what target the user is gazing at. The method was described in detail in Section 3.1.1. They used an EEG acquisition device with a sampling rate of 512 Hz, an LCD monitor with a 120 Hz refresh rate, and used Oz as the sensor location. The method operates on two sets of data, named $SData$ and $CData$. $SData$ is the data collected over the previous 2 seconds, while $CData$ contains the last three sets of $SData$ (6 seconds of data). Before FFT is performed on the data sets, an analog bandpass filter is applied between 5-30 Hz, and autocorrelation is applied to the data to increase the SNR. The data is then zero-padded to achieve an FFT resolution of 0.1 Hz. By using Equation 3.1, the data has to be zero-padded to a total of 5120 samples. The reason for zero-padding is that the targets use the frequencies 6 Hz, 6.5 Hz, 7Hz, 7.5 Hz, 8.2 Hz, 9.3 Hz, 10 Hz, and 11 Hz, nine targets in total. As explained in Section 3.1.1, the frequencies of interest have to be a multiple of the FFT resolution. After the FFT has been applied, the score for each class, C_x, is calculated as the sum of power amplitudes, $|Y|$, within ± 0.1 Hz of the fundamental frequency, $H1$, and the second harmonic, $H2$:

$$C_x = \sum_{H1-0.1}^{H1+0.1} |Y| + \sum_{H2-0.1}^{H2+0.1} |Y|, \quad x = 1, ..., n. \tag{4.1}$$

The different C_x classes are then normalized with regards to one another, with the dominating class having the value of 1. If at least one of the following three criteria are met, the dominating class is chosen:

- The second greatest value in $SData < 0.35$.
- The second greatest value in $CData < 0.45$.
- The same class, C_x, is dominating in four consecutive iterations.

Implementation

Some changes had to be made when implementing this method with Emotiv EPOC. Emotiv EPOC only offers a 128 Hz sampling rate compared to the 512 Hz used in this method, and an LCD monitor with a refresh rate of 60 Hz is used for all experiments conducted in this thesis. Emotiv EPOC does not offer a sensor at the Oz location. The latter problem was solved by using two other sensor placements: O1 and O2. The PSD was calculated for the EEG data from both O1 and O2, and the average of the two was used. Figure 4.7 shows the O1 and O2 placements relative to Oz.

The implemented PSD algorithm uses NumPy's FFT implementation[9]. NumPy's FFT approximates the DFT, and defines the DFT as

$$A_k = \sum_{m=0}^{n-1} a_m \exp\left\{-2\pi i \frac{mk}{n}\right\} \quad k = 0, ..., n-1. \tag{4.2}$$

a_m is represented by the complex exponential $a_m = \exp\{2\pi i f m \Delta t\}$ where Δt is the sampling interval. The FFT function takes as input an array of length N_{fft}, and returns an array that has the same number of elements and contains complex numbers. The first element of the array contains the mean of the signal. The next elements up to $N_{fft}/2+1$ contain all the positive frequency terms, which holds the relevant information for a time-domain signal such as EEG. These frequencies are spaced out by the FFT resolution, as described by Equation 3.1 in Section 3.1.1: f_s/N_{fft} ,where f_s is the sampling rate. Each element in the FFT array is of the complex form $A_j = a + ib$. To get the PSD from the FFT, the absolute value of the first $N_{fft}/2 + 1$ elements in the FFT array are calculated and squared:

$$PSD_j = abs(A_j)^2 = a^2 + b^2, \quad j = 1, ..., N_{fft}/2 + 1, \tag{4.3}$$

where j refers to the j-th element of the FFT array, and $abs()$ is the absolute value of a complex number, defined as $abs(a + ib) = \sqrt{a^2 + b^2}$.

$SData$ contains two seconds of data, so the frequency resolution ΔR_f is $1/2 = 0.5$ Hz. Vilic et al. zero-padded the EEG signal to achieve an FFT resolution of 0.1 Hz, due to the targets using frequencies that were a multiple of 0.1 Hz. However,

[9] `http://docs.scipy.org/doc/numpy/reference/routines.fft.html`

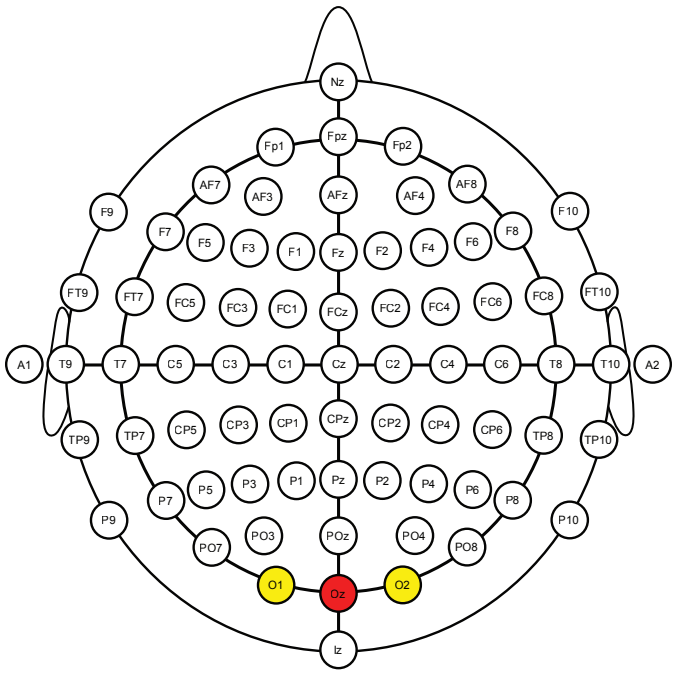

Figure 4.7: The EEG channel used for the original M3 method compared to the ones used by Emotiv EPOC. The ones used by Emotiv EPOC are yellow, while the one used by the original method is red. Adapted from Marius 't Hart - `http://www.beteredingen.nl`. Used under CC BY: `http://creativecommons.org/licenses/by-sa/3.0/nl/deed.en_GB`.

for the implementation in this thesis, the choice was made to only use whole or half frequencies, e.g. 12.0 Hz and 12.5 Hz, so zero-padding is not needed. Before calculating the PSD, a digital bandpass filter between 5-45 Hz is applied. Since the method uses the fundamental frequency and its second harmonic, the bandpass filter allows the use of frequencies from 5-22.5 Hz for the flickering targets.

Vilic et al. performed autocorrelation on the data prior to taking the FFT to obtain the PSD. Calculating the sample-by-sample correlation of a signal with itself to obtain the autocorrelation, as described in Section 3.1.1, can be computationally expensive for large sets of data. Instead, the autocorrelation can be computed with an FFT-based approach. By taking the FFT of the EEG data, multiplying each element in the FFT array by its complex conjugate and then taking the inverse FFT, results in the cyclic autocorrelation of the data.

The complex conjugate of a complex number $c = a + ib$ is $\bar{c} = a - ib$. Multiplying these two numbers gives $c \cdot \bar{c} = a^2 + b^2$. Looking at Equation 4.3 for calculating the PSD, multiplication with the complex conjugate yields the same result as the PSD calculation. This is known as the Wiener-Khinchin theorem, and it states that the PSD of a signal is the Fourier transform of the corresponding autocorrelation function [54]. In other words, calculating the PSD directly yields the same result as running autocorrelation on the data and then applying the FFT.

The implemented method in this thesis takes the EEG data and applies a Hanning window to it, for the reasons described in Section 3.1.1. A Hanning window was chosen because it is what Emotiv uses[10]. Then the FFT of the signal is used to calculate the PSD as described by Equation 4.3. After the PSD has been computed, each target (class) C_x is calculated as the sum of the fundamental frequency, $H1$, and second harmonic $H2$ of the target frequency in the PSD:

$$C_x = |H1| + |H2|, \tag{4.4}$$

The implemented method uses the same three selection criteria as the original method by Vilic et al.

4.3.4 M4 using c-VEP

Bin et al. [2] presented a single channel c-VEP method. This method was described in detail in Section 3.1.2. The method uses a training period where the user focuses on a reference target for N trials. The average over the N trials, R, is calculated from the EEG data and is used as the template for the reference target. The reference target uses a binary m-sequence of length 63 for its stimuli on a 60 Hz LCD monitor. The original m-sequence is shifted by two bits for each additional target to create a different stimulus pattern. The time delay between two consecutive targets is $\tau = k/60$, where k is the number of bits the m-sequence is shifted by. For $k = 2$, $\tau = 0.033s$. R is then the template for the reference target. The reference

[10] `http://www.emotiv.com/ideas/forum/forum4/topic3829/`

template, R, is shifted in time according to τ to create templates for the other targets. Classification is performed for a new trial by calculating the correlation coefficient between the trial and the templates for all targets. The template that has the highest correlation coefficient with the trial is selected as the fixation target. Bin et al. used Oz as the single channel.

Implementation

Since Emotiv EPOC does not have the Oz sensor location that was used in the original method, O1 was used in this implementation. A binary m-sequence was generated using a code snippet written by GitHub user *mubeta06*[11]. The m-sequence of length 63 is shown as bit sequence T1 in Figure 3.7. The correlation coefficients between a new trial and the templates are calculated using NumPy's *corrcoef*[12] method.

Testing the M4 implementation on one of the authors gave poor performance. This author had good sensor connectivity with Emotiv EPOC, and obtained satisfying results with the three other methods, M1, M2 and M3. Results from testing M4 showed no better prediction than predicting a random target, with the m-sequence having lengths of 63 bits (1.05 seconds) and 127 bits (2.12 seconds). A shift size of 8 bits when the m-sequence had length 63, and 15 bits when of length 127, were used for the different targets. The reference template, generated by the process described above, has to be shifted by n samples, depending on τ. The number of samples, T, that the reference template has to be shifted by can be calculated by

$$T = \tau \cdot f_s \tag{4.5}$$

where f_s is the sampling rate of the EEG acquisition device. For Emotiv EPOC, it is 128 Hz. 8 bits results in a template shift of 17.07 samples, while 15 bits results in a template shift of exactly 32 samples. Minimizing the shifting error results in more accurate templates.

A debugging process found nothing wrong with the implementation to explain the poor performance of the method. The next step was to determine if the recorded EEG data contained a consistent c-VEP response. Data was recorded from one of the authors while looking at a reference target for 150 trials, with an m-sequence of length 127. The average of the trials was computed as the template, and the pairwise correlation between each trial and the template was calculated. The average correlation for the trials and the template was 0.027. The correlation between each trial and the template is shown in Figure 4.8. The stimulus was the same for all trials and should have resulted in a good correlation. This was not the case for the implemented method. There are two likely causes for it: (1) too much noise in the signal, (2) lack of accurate timing information for the EEG data.

[11] `https://github.com/mubeta06/python/blob/master/signal_processing/sp/mls.py`
[12] `http://docs.scipy.org/doc/numpy/reference/generated/numpy.corrcoef.html`

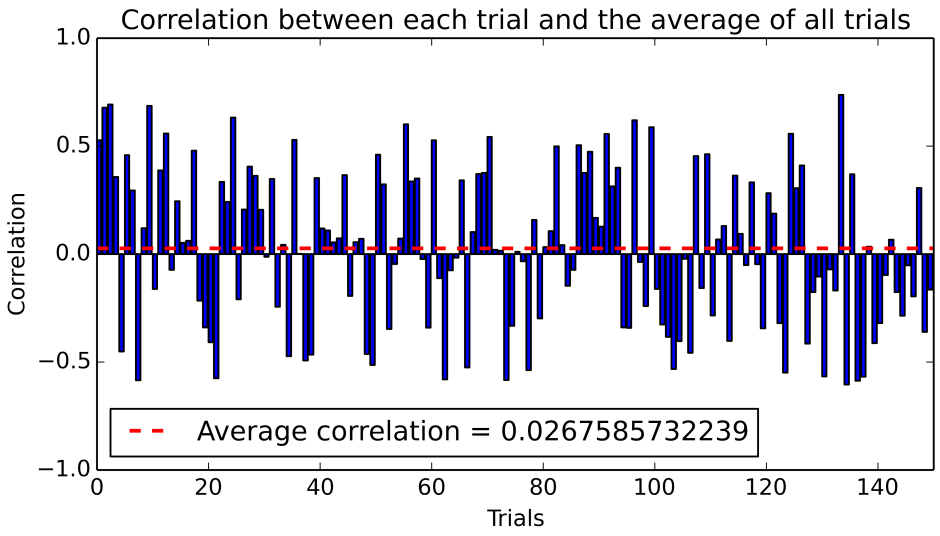

Figure 4.8: The correlations between each trial and the average of all trials. The reference target was gazed upon for 150 trials.

The synchronization problem

The most likely problem was identified as a synchronization problem, where the acquisition of samples from the headset was not synchronized with the frames drawn on the monitor. In the following, the synchronization problem is described, and the attempted method to handle the problem is presented.

Emotiv EPOC acquires samples every 1/128 s, but as described in Section 4.1.1, the data is not put in the headset buffer for reading until four samples are recorded. Using Emotiv EPOC in a c-VEP system results in some challenges. When the first stimulus frame is drawn on the monitor for a new trial, a time delay will occur before the group of four samples is available in the data buffer. This delay, t_s, varies between between 0 seconds and 4/128 seconds. The varying delay results in different c-VEP trials having different starting points for their first samples. A variation of up to four samples is equal to 1/32 seconds, and represents a shift of almost two frames (or bits), $2/60s = 1/30$ seconds. This is a significant variation, as Bin et al. [2] shifted their stimulation sequence by two bits. Having this kind of variation error will break a c-VEP system. Another problem with the buffering, is that some of the four samples might be recorded from before the first frame in a stimulus sequence is displayed, which means a trial can contain invalid data. Invalid data contained in a trial causes problems when shifting the data to obtain templates for other targets.

A procedure was implemented to ensure that only relevant samples were included

in the recorded EEG data:

1. The time t_f was recorded immediately after the first stimulus frame for a new stimulus cycle was set. In Section 4.1.2, it was described how the *visual.Window.flip()* function returns control to the program when the front and back buffers have been swapped and the frame has been drawn.

2. Four samples were fetched from the headset. The time that the samples became available, t_r, was recorded.

3. $t_d = t_r - t_f$ was calculated. t_d would have the range $0 < t_d < 4/128$.

4. An estimate was calculated of how many samples that should have been recorded in time t_d: $x = \lfloor \frac{t_d}{(1/128)} \rfloor$.

5. The x last samples from the group of four samples were extracted. The rest of the samples were assumed to be recorded before the frame was drawn and were discarded.

The above procedure was used after the first frame so that the c-VEP cycle uses the correct EEG data. For the rest of the cycle, all samples gathered from the headset were recorded. The number of samples in a cycle recording was calculated by ⌈(bitStringSize / 60) * 128⌉.

Even though the synchronization procedure looked promising, running another c-VEP test with 50 trials still resulted in poor performance. The average correlation between the trials and the generated template was 0.197, and is shown in Figure 4.9. A possible problem with the procedure is that the current time was acquired through the Python method *time.time()* which could be inaccurate, causing t_d to be inaccurate. However, a more likely explanation of the poor c-VEP data, is the unknown time that the Emotiv EPOC uses to preprocess recorded data and write them to the headset buffer. As Table 2.1 shows, the Emotiv headset uses a digital 5th order Sinc filter and digital notch filters at 50 Hz and 60 Hz. The time to perform these operations and any other unknown operations on the data, can significantly delay the data arriving in the buffer. Without knowing the duration of this processing it would be difficult, or even impossible, to find which samples were recorded during the stimulus sequence.

As seen, the implemented c-VEP method from Bin et al. [2] had synchronization challenges using Emotiv EPOC, resulting in poor c-VEP recordings. It was therefore decided not to implement the more advanced c-VEP method by Spüler et al. [38], as this method would have the same basic problem of working with inadequate c-VEP data. The c-VEP methods were also removed from the list of techniques that would be tested by the test subjects.

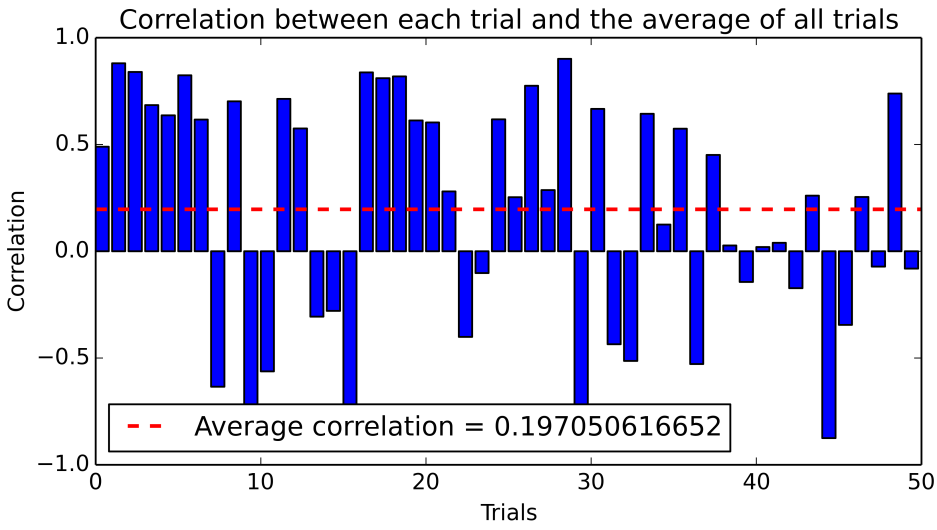

Figure 4.9: The correlations between each trial and the average of all trials, after the synchronization procedure was applied. The reference target was gazed upon for 50 trials.

Chapter 5

Experiments and Results

This chapter describes the experiments conducted and their results. Section 5.1 presents the experimental plan, including the tests that were performed and the number of test subjects. Section 5.2 describes the experimental setup, such as the visual stimulator configuration. Finally, Section 5.3 presents the results obtained from the experiments, and describes how the ITR was calculated.

5.1 Experimental Plan

In Section 3.4.3, four methods were selected to be implemented and tested. These methods were:

- M1 by Liu et al. [29]
- M2 by Cao et al. [6]
- M3 by Vilic et al. [42]
- M4 by Bin et al. [2]

The c-VEP method by Bin et al. was not tested due to the difficulties described in Section 4.3.4. The test plan is shown in full in Table 5.1.

The first two tests were performed to determine what color the stimuli should be for the remaining six tests. During preliminary tests conducted to verify the implementation of the different methods, it was found that one of the authors responded notably better when the stimuli was presented in red color as opposed to white or gray color. This disagrees with the findings by Cao et al. [5] in Section 3.3. This discrepancy was the reason behind including a color test. The method by Liu et al. [29] was used to determine which color was best for the test subject. The choice behind using this technique was that Liu et al. [29] achieved good results when using Emotiv EPOC.

Test ID	Method	Window length	Thresholds	Color	Feature extraction
1	M1	3 seconds	-	White	CCA
2	M1	3 seconds	-	Red	CCA
3	M1	2 seconds	-	-	CCA
4	M2	3 second	0.2	-	CCA
5	M2	2 seconds	0.2	-	CCA
6	M2	1 seconds	0.2	-	CCA
7	M3	-	0.35, 0.45	-	PSDA
8	M3	-	0.5, 0.6	-	PSDA

Table 5.1: The test plan using the three implemented SSVEP methods.

The original authors selected certain parameters for use in their methods. These parameters were found empirically through trial and error based on the hardware used and the software implementation. The combination of hardware and software differs from the hardware and software used in this thesis, so it was probable that different parameters for the experiments ran in this thesis could give better results than the original parameters. Liu et al. [29] used a window length of three seconds. While it could be expected that using the same window length would yield similar results since the method used Emotiv EPOC, a different window length could achieve better results due to differences in the software implementation. That is why a test with a two second window was included for this method. Cao et al. [6] used a medical grade EEG acquisition device and a window length of one second with a threshold of 0.2. Emotiv EPOC has been shown to have a significantly worse SNR than medical grade equipment (see Section 2.3), so a window length longer than one second could yield better results. This is why tests with both a two and three second window were included. The threshold of 0.2 remained fixed for all tests. Vilic et al. [42] used a window length of two seconds, with thresholds 0.35 and 0.45 for $SData$ and $CData$ respectively, with a medical grade EEG acquisition device. More relaxed thresholds were tested in addition to the original thresholds to see if they would give better results for noisier EEG data. These thresholds were 0.5 for $SData$ and 0.6 for $CData$. The window length of two seconds for $SData$ and six seconds for $CData$ remained unchanged.

Eight test subjects were asked to test the different methods. These subjects were aged from 22 to 27, six male and two female. This is summarized in Table 5.2. All subjects had normal or corrected-to-normal vision. The experimental process can be split into three steps:

1. Introduction and setup

During the introduction, the test subject had time to get familiar with the system and how it works. It can be difficult to adjust Emotiv EPOC to achieve the best signal quality for VEPs. The M1 method by Liu et al. [29] was used with a three second window over six trials to adjust the sensor locations. The sensor placements that gave the highest correlation coefficients,

Subject	Age	Gender
S1	27	M
S2	25	M
S3	22	F
S4	26	M
S5	27	M
S6	25	F
S7	26	M
S8	25	M

Table 5.2: List over the test subjects.

with the assumption that a higher correlation means better signal quality, were chosen.

2. Color testing
After good signal quality was achieved, the test subject performed tests 1 and 2 in Table 5.1 to determine which color elicited the strongest SSVEPs. The best color was determined by the highest ITR. This color was then used for the remaining tests (3-8).

3. Method testing
Tests 3-8 in Table 5.1 were performed.

5.2 Experimental Setup

The experiments were conducted on a 24" LCD monitor with a 60 Hz refresh rate and a resolution of 1920x1200. The test subject was situated in a comfortable chair 60 cm in front of the monitor. The SSVEP was elicited by six targets (boxes), each flickering at a different frequency. These frequencies were 12 Hz, 12.5 Hz, 13 Hz, 13.5 Hz, 14 Hz and 14.5 Hz for all tests. These frequencies were selected based on the conclusion by Liu et al. [29] that subjects with strong alpha waves can have worse performance when target frequencies overlap with the alpha band (frequencies between 7.5-12.5 Hz). Each of the 8 tests described in Table 5.1 were performed for 42 trials. Each trial started with the targets flickering for some time and then a break period with no targets flickering. During the break period the subject received no stimuli, and no EEG was recorded. During the break period an arrow would appear on top of the next reference target, so that the subjects would know where to gaze in the next trial. The arrow would appear green if the classification of the previous trial was correct, and yellow otherwise. A counter was placed in the middle of the screen to show the test subject how many trials were left in the current test. This is shown in Figure 5.1. With 42 trials, each target was selected as the reference target seven times. The order of the boxes was randomized for each test. Users were allowed to take five minute breaks between tests if they felt tired.

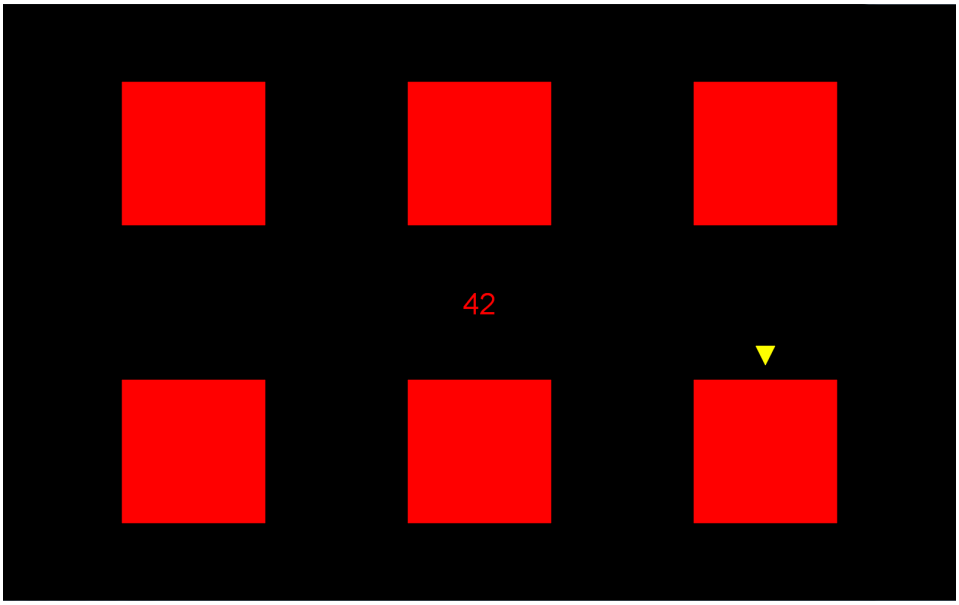

Figure 5.1: The visual stimulator for tests conducted, here shown with red targets. The yellow arrow shows what target the user should gaze at next.

Subject	Age	Gender
S1	27	M
S3	22	F
S6	25	F
S8	25	M

Table 5.3: List over the test subjects.

Out of the eight subjects, four subjects were unable to get a good fit for the headset. As discussed in Section 4.2, a poor fit for the headset results in either extreme sensitivity to external noise or poor sensor coverage of the visual cortex. Test subjects S1, S3, S6 and S8 performed all the tests, while S2, S4, S5 and S7 were not able to get satisfactory signals due to the headset not fitting properly. Since these subjects would not provide any meaningful data, they were excluded from further analysis. Table 5.3 shows the updated list of test subjects. All experiments were conducted in a quiet room, so that the test subjects were not distracted by activity around them.

The experimental setup fulfills suggestions 2, 4, 6 and 7 for standardized ITR calculations by Yuan et al. [53], presented in Section 2.2.3.

5.3 Experimental Results

The results of the tests are presented here using ITR instead of PBR since PBR is not yet widely used when reporting BCI results. There are three more criteria (1,3 and 5) that have to be met to fulfill the seven criteria for standardized ITR calculations, according to Yuan et al. [53]. Standardizing the performance measurement will allow others to understand how the results were calculated, and they can then better compare their own work to the work presented in this thesis. The three remaining criteria are:

- When reporting the ITR, N, P and T in Equation 2.4 and Equation 2.5 should be explicitly identified.

N is the number of targets, which is set to six for all tests. P is the accuracy of the classification algorithm, and T is the detection time in seconds. The detection time is the time it takes for the method to identify what the user is looking at. For the tests involving method M2, T is a constant. For M1 and M3, T varies.

- When reporting the ITR, authors should explain all of the factors in the ITR calculation, such as whether t_1 is included in the calculations. t_1 is the time it takes the user to shift attention to a new target.

t_1 in this case is the one second break between each trial where the user can shift his or her gaze to the new reference target. t_1 is **not** included in the ITR calculations for any test .

- Authors should include an ITR estimation that does not include error correction or other methods to increase effective throughput. If a system does employ error correction, authors should adequately describe the methods and results and, if desired, include a modified ITR as well.

Error correction is not employed for any of the performed tests.

5.3.1 Test results

Table 5.4 shows the average for the four test subjects for each test performed. The average ITR is calculated from the ITR of each test subject, not from the average accuracy and average detection time. Test 1 and 2 use the same method with the same parameters, with only the color of the targets being changed. The remaining tests from 3 through 8 use the color for which the test subject had the best performance with from the first two tests. To get an accurate estimation of the performance of the method used in the first two tests, the final row of the table shows the results when the best result from tests 1 and 2 was selected for each subject, denoted as the Test ID “max(1,2)”. Figure 5.2 presents Table 5.4 in graphically. Table 5.5 presents the complete results of all tests performed on all four subjects. Each sub-table presents the results for one method.

Test ID	Method	Accuracy (%)	Detection time (sec)	ITR (bits/min)
1	M1	79.76 ± 15.57	4.16 ± 0.16	22.16 ± 10.84
2	M1	82.14 ± 15.93	4.10 ± 0.16	24.31 ± 11.55
3	M1	86.91 ± 8.67	3.33 ± 0.19	32.36 ± 8.94
4	M2	89.88 ± 2.60	4.00 ± 0.00	28.24 ± 2.09
5	M2	85.12 ± 4.58	3.00 ± 0.00	32.92 ± 4.72
6	M2	55.96 ± 7.24	2.00 ± 0.00	17.64 ± 5.80
7	M3	92.26 ± 6.60	7.15 ± 2.01	18.55 ± 5.41
8	M3	89.29 ± 6.41	5.14 ± 0.98	23.78 ± 7.15
max(1,2)	M1	92.86 ± 7.14	4.02 ± 0.14	31.72 ± 7.19

Table 5.4: The average results of all tests performed.

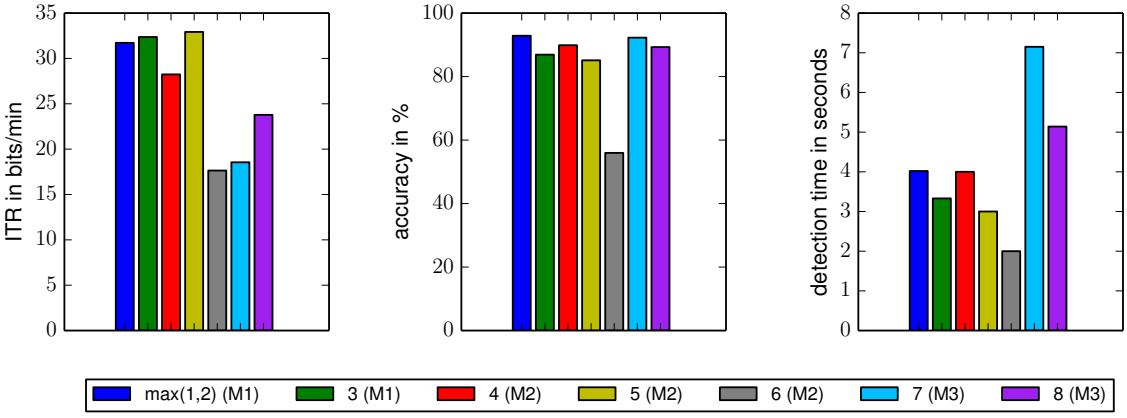

Figure 5.2: The average results of the tests. The numbers in the legend correspond to the test ID of Table 5.4, and the parenthesis shows the method that was used. The figure shows the ITR, accuracy and detection time for each method.

Test ID 1	S1	S3	S6	S8	**Average**
Accuracy (%)	66.67	95.24	95.24	61.91	**79.76 ± 15.57**
Detection time (sec)	4.35	4.02	3.98	4.30	**4.16 ± 0.16**
ITR (bits/min)	12.33	32.78	33.17	10.34	**22.16 ± 10.84**

Test ID 2	S1	S3	S6	S8	**Average**
Accuracy (%)	100.00	57.14	90.48	80.95	**82.14 ± 15.93**
Detection time (sec)	3.83	4.21	4.14	4.23	**4.10 ± 0.16**
ITR (bits/min)	40.46	8.62	27.70	20.45	**24.31 ± 11.55**

Test ID 3	S1	S3	S6	S8	**Average**
Accuracy (%)	97.62	73.81	90.48	85.71	**86.91 ± 8.67**
Detection time (sec)	3.27	3.66	3.20	3.19	**3.33 ± 0.19**
ITR (bits/min)	43.49	18.83	35.79	31.31	**32.36 ± 8.94**

Test ID 4	S1	S3	S6	S8	**Average**
Accuracy (%)	92.86	90.48	85.71	90.48	**89.88 ± 2.60**
Detection time (sec)	4.00	4.0	4.00	4.00	**4.00 ± 0.00**
ITR (bits/min)	30.72	28.65	24.92	28.65	**28.24 ± 2.09**

Test ID 5	S1	S3	S6	S8	**Average**
Accuracy (%)	83.33	80.95	83.33	92.86	**85.12 ± 4.58**
Detection time (sec)	3.00	3.00	3.00	3.00	**3.00 ± 0.00**
ITR (bits/min)	30.96	28.81	30.96	40.96	**32.92 ± 4.72**

Test ID 6	S1	S3	S6	S8	**Average**
Accuracy (%)	47.62	64.29	50.00	61.91	**55.96 ± 7.24**
Detection time (sec)	2.00	2.00	2.00	2.00	**2.00 ± 0.00**
ITR (bits/min)	11.11	24.46	12.72	22.25	**17.64 ± 5.80**

Test ID 7	S1	S3	S6	S8	**Average**
Accuracy (%)	100.00	97.62	85.71	85.71	**92.26 ± 6.60**
Detection time (sec)	6.05	9.86	4.57	8.10	**7.15 ± 2.01**
ITR (bits/min)	25.65	14.41	21.81	12.32	**18.55 ± 5.41**

Test ID 8	S1	S3	S6	S8	**Average**
Accuracy (%)	100.0	83.33	88.10	85.71	**89.29 ± 6.41**
Detection time (sec)	5.57	6.24	3.57	5.19	**5.14 ± 0.98**
ITR (bits/min)	27.84	14.89	33.17	19.21	**23.78 ± 7.15**

Table 5.5: Results of the tests in Table 5.1

Chapter 6

Evaluation and Discussion

This chapter evaluates the results of the experiments described in Chapter 5 and discusses the work done in this thesis. Section 6.1 evaluates the work done in this thesis in relation to the research questions defined in Chapter 1, and evaluates the test results. Section 6.2 discusses the merits and the limitations of the work, as well as challenges faced throughout the project. Section 6.3 summarizes the main contributions to the field and their significance. Finally, Section 6.4 suggests what may be done to extend the work performed in this thesis.

6.1 Evaluation

Section 1.2 states that the goal for this thesis is "to compare VEP-based BCI methods while using the low-cost EEG acquisition device Emotiv EPOC". In this context, two relevant research questions were asked. To answer both research questions, an SLR was conducted. The SLR protocol is described in Appendix A, and the results found are described in detail in Chapter 3. A subset of these methods were selected (Section 3.4.3), implemented (Section 4.3) and tested (Chapter 5) to achieve the goal of the thesis.

Research question 1 Which techniques can be used to classify the VEPs present in the EEG data?

Through the SLR, current techniques used for VEP-based BCIs were identified. The identified SSVEP techniques were PSDA [42, 15], CCA [6, 29] and MEC [43]. In addition, methods that modify traditional SSVEP-based BCIs by use of adaptive time-window length [14] and amplitude-modulated visual stimulation [10] were found. Two c-VEP techniques were identified. One method uses OCSVM for classification [38], and the other uses template matching for classification [2]. Four methods were implemented. Three of these methods are SSVEP methods, and one is a c-VEP method. These are shown, along with their identifier, in Table 6.1

Research question 2 How is a system to compare VEP-based BCI methods implemented and configured?

Identifier	Original author	Feature extraction	VEP paradigm
M1	Liu et al. [29]	CCA	SSVEP
M2	Cao et al. [6]	CCA	SSVEP
M3	Vilic et al. [42]	PSDA	SSVEP
M4	Bin et al. [2]	Template matching	c-VEP

Table 6.1: The VEP methods that were chosen for implementation and testing.

A system used in a VEP-based BCI consists of two parts: the visual stimulator and the EEG acquisition module. The visual stimulator in this thesis was limited to using an LCD monitor with a 60 Hz refresh rate. One of the issues when displaying stimulus that flickers at a given frequency is that the monitor refresh rate has to be divisible by the frequency. Through the SLR, an approach was found that produces a stimulus equivalent to any frequency [46]. The SLR also revealed that some people have high activity in the alpha band (7.5-12.5 Hz) causing frequencies in this range to have a superficially high magnitude in the PSD [29]. Furthermore, a 30 Hz stimulus is the highest frequency that a monitor with a 60 Hz refresh rate can display. As a result of these findings, the visual stimulator was configured to display stimuli from 12-14.5 Hz.

The second part of the system is concerned with retrieving the EEG data. Due to limitations of Emotiv EPOC, a c-VEP implementation with the M4 method was not possible. This was the result of a synchronization issue between the stimulus and the recorded EEG data.

6.1.1 Comparison of SSVEP techniques for Emotiv EPOC

Table 5.4 shows the average results for all the techniques used, with different parameters. It is seen from the results that the M2 method proposed by Cao et al. [6], has a lower accuracy than M1 and M3. The highest average accuracy for M2, 89.88%, was achieved when the window length was set to three seconds, with a total gaze length of four seconds. After four seconds, the technique is forced to output either a prediction or unclassified if the result does not exceed a predefined threshold. M1, the method proposed by Liu et al. [29], achieved an average accuracy of 92.86% when using a three second window, while M3, the method proposed by Vilic et al. [42], achieved an average accuracy of 92.26% when using thresholds 0.35 for $SData$ and 0.45 for $CData$. Both of these two methods continue to collect EEG data until certain conditions are met for classification. This explains why these methods have a higher accuracy than M2.

The highest average ITR observed, 32.92 bits/min, came from the M2 method, using a two second window and a total gaze time of three seconds. Both M1 variants were close. With a window length of three seconds, the M1 method achieved an ITR of 31.72 bits/min. With a two second window, the ITR was 32.36 bits/min. While the accuracy was higher for both M1 variants, M2 had constant detection

time set to three seconds, which was lower than what both M1 variants achieved. This shows that a higher ITR can be achieved even with a lower accuracy if the detection time is lower as well.

The highest individual ITR observed was 43.49 bits/min. This result was achieved by subject S1 using the M1 method with a window length of two seconds.

M3 had the lowest average ITR of all the methods. With the original thresholds of 0.35 for $SData$ and 0.45 for $CData$, it achieved an accuracy of 92.26%, detection time of 7.15 seconds and an ITR of 18.55 bits/min. With more relaxed thresholds of 0.5 for $SData$ and 0.6 for $CData$, the method had an accuracy of 89.29%, detection time of 5.14 seconds and an ITR of 23.78 bits/min. The low ITR can be attributed to the long detection times. It is interesting to see how the change in thresholds resulted in an ITR increase of 5.23 bits/min, 2.01 second decrease in detection times and a decrease of only 3 percentage points in accuracy. This suggests that the original thresholds were too rigorous for a headset as noisy as Emotiv EPOC. M3 was the only method that used PSDA, the two other methods used CCA. The findings presented here reaffirms that CCA outperforms PSDA also for Emotiv EPOC in terms of ITRs. The test results show that an increase in window length results in a higher accuracy for all the implemented methods. Emotiv EPOC has been proven to be capable of detecting SSVEPs, with a high enough SNR to reach accuracies of over 90%.

6.1.2 Comparison to the original methods

When comparing BCI methods by using ITR, knowing how the ITR was calculated is crucial. The articles that presented the three methods implemented in this thesis did not follow the standards outlined in Section 2.2.3, and assumptions would have to be made in order to deduce exactly how the ITR was calculated. In addition, the *number of targets* is one of the variables used in calculating the ITR. The visual stimulator implemented here uses six targets, and only M1 uses the same number of targets out of the three methods in the original articles. Under these circumstances, a direct comparison of the reported ITRs from the articles and the ITRs achieved in this thesis yields little information. No studies were found that addressed the issue of how the number of targets in an SSVEP system affects the accuracy or detection time of a classification method. It is possible that with a larger number of targets, the area for each target is smaller, and there is more interference from the non-reference targets, which can elicit less prominent VEPs. Accuracy and detection times are however reported in all three articles, and is the basis for comparison for the rest of this section. Table 6.2 shows how the different methods compare when using the original parameters.

The implementation of the M1 method is the one that most closely resembles the method presented in the original article [29]. Both implementations use six targets and collect EEG data with the Emotiv EPOC headset. Different target frequencies were used. The original paper used frequencies 6.67 Hz, 7.5 Hz, 8.57

Method	Average accuracy (%)	Average DT (sec)	Window length	Thresholds
M1 Liu et al. [29]	95.83 ± 3.59	5.25 ± 2.14	3 seconds	N/A
M1 (implemented)	92.86 ± 7.14	4.02 ± 0.14	3 seconds	N/A
M2 Cao et al. [6]	98.78 ± 1.62	2.00 ± 0.00	1 second	0.2
M2 (implemented)	55.96 ± 7.24	2.00 ± 0.00	1 second	0.2
M3 Vilic et al. [42]	90.81 ± 4.11	6.62 ± 1.03	2 seconds	0.35, 0.45
M3 (implemented)	92.26 ± 6.60	7.15 ± 2.01	2 seconds	0.35, 0.45

Table 6.2: A performance comparison of the implemented methods and the original methods from the articles. DT stands for detection time.

Hz, 10 Hz, 12 Hz and 15 Hz, while 12 Hz, 12.5 Hz, 13 Hz, 13.5 Hz, 14 Hz and 14.5 Hz were used in the experiments presented in Chapter 5. In the original paper, the method uses a three second window. It achieved an average accuracy of 95.83 %, with an average detection time of 5.25 seconds. The M1 implementation in this thesis achieved an average accuracy of 92.86 % and an average detection time of 4.02 seconds, when using the same window length.

It is interesting that the average detection time of all subjects for the method implemented in this thesis was lower than the lowest individual average detection time reported by Liu et al. [29]. Some possible reasons for this may include variance of susceptibility to SSVEP stimuli for the test subjects, or a visual stimulator implementation that provides more accurate stimuli. It is also possible that the authors of the paper neglected to mention an important implementation detail, or that the authors of this thesis misunderstood the information provided in the paper.

Table 6.2 also shows that the M2 method only achieved an average accuracy of 55.96% using a window length of one second. The original method, using the same window length, had an average accuracy of 98.78%. The main difference between the original implementation and the one tested in this thesis is the headset used. Emotiv EPOC has a lower sampling rate, a lower SNR and has fewer sensors covering the occipital lobe and the visual cortex than the one used by Cao et al. [6], and can account for the difference. The accuracy increases as the window length increases, and with a three second window, M2 achieved an average accuracy of 89.88%.

The implementation of the M3 method achieved similar results to the original method presented by Vilic et al. [42]. Vilic et al. obtained an average accuracy of 90.81% and average detection time of 6.62 seconds. The implementation of M3 had an accuracy of 92.26% and detection time of 7.15 seconds. Vilic et al. used only a single sensor placement, Oz, while the implemented technique used the average of O1 and O2. Since Oz is positioned in the middle of the visual cortex, it gives better VEPs than O1 and O2. In the original paper [42], a medical grade EEG headset was used. That Emotiv EPOC achieved similar results is surprising. When changing the thresholds to 0.5 for $SData$ and 0.6 for $CData$, the accuracy

was 89.29%, 1.52 percentage points lower than the original. The detection time was 5.14 seconds, which is 1.48 seconds faster than the original.

6.2 Discussion

The work of Liu et al. [29] showed how Emotiv EPOC compared to a more expensive medical grade headset. While comparing the two EEG acquisition devices, Liu et al. did not explore the performance of Emotiv EPOC in greater detail by testing different methods. The work of this thesis has extended Liu et al.'s work by further comparing VEP-based methods on the same low-cost hardware. The results show that Emotiv EPOC achieves satisfactory results with the implemented SSVEP techniques. Hopefully, this thesis and further research on low-cost EEG acquisition devices can contribute to an increase in the attractiveness of VEP systems for commercial use.

An important finding in this thesis was the difficulties using c-VEP with Emotiv EPOC. The implementation revealed synchronization issues between the visual stimulator and Emotiv EPOC. Using the DLLs from Emotiv, it is not possible to know which samples were recorded right after the first frame in a c-VEP cycle. As long as this synchronization issue remains, it is not possible to use c-VEP with Emotiv EPOC.

The implementation and testing of the VEP system also revealed interesting information about the color of the flickering boxes. Cao et al. [5] concluded that white and gray color on the stimuli results in a more intense SSVEP. The results from this thesis showed that two of the four test subjects performed significantly better with red stimulus compared to white stimulus and thereby contradicts the findings by Cao et al. [5]. These results are based on a limited sample size, and a larger sample size is required to make a conclusion on the subject. This is one of the **limitations** of the work performed in this thesis. Another limitation is how the tests are performed. As described in Section 5.2, each test subject completed 42 trials for each method. This gave a good indication of the performance of the method, but for more precise statistics, more trials are needed. Only half of the test subjects were able to use Emotiv EPOC, making calibration and testing a time consuming process. For this thesis, the authors considered the number of trials to be satisfactory.

In the SLR, ITR was included as a required search term for papers describing VEP techniques. This was done to ensure that the resulting articles used ITR as a metric for ease of comparison. However, this search term also resulted in a more narrow search, and articles describing important VEP techniques in the field may have been missed. The SLR resulted in a chapter describing many VEP techniques. Only a subset of these were selected for implementation and testing. Testing the unimplemented methods may give valuable information about the performance of Emotiv EPOC, but that was outside of the scope of this thesis due to

time constraints.

6.2.1 Challenges

As touched upon in the introduction of this thesis, building a complete VEP system requires knowledge from several disciplines, including mathematics, statistics, signal processing, neuroscience and artificial intelligence. For the authors of this thesis it has been challenging to build a VEP-based BCI from the ground up being only computer science students. Many tasks required a deep understanding of Fourier analysis, canonical correlation and signal processing (e.g autocorrelation), and it has been time consuming to read up on areas like these. In addition to the technical knowledge, it has been necessary to learn about the vast field of VEPs and BCIs. In this process, the SLR has been helpful in getting an overview of the field and finding state of the art techniques within these areas.

It has not only been challenging to obtain the required knowledge, but building a well functioning VEP-based BCI from the ground up involves a considerable amount of programming. It has been a time consuming process to build the architecture of this system with many concurrent threads, a visual stimulator program and calculations on a server. The implementation challenges have been many, one of them being how to process data from Emotiv EPOC. As described in Section 4.2, the EEG values from the sensors of Emotiv EPOC have a DC offset causing the mid-range point of the data to be at 4096 units. The authors think that this information should have been available through the manual that comes with the headset. Instead, a search through the Emotiv forums were necessary, where one of the representatives from Emotiv presented these facts.

6.3 Contributions

This thesis has given a comprehensive introduction to the field of VEP. For newcomers to the field, the background theory can be helpful. The presentation of the related work gives a systematic and objective overview over existing VEP solutions, and can be useful for researchers wanting an update on the field. Furthermore, details of how a VEP-based BCI can be built have been presented, and can be useful for everyone wanting to build such a system from the ground up. In most of the articles pertaining to VEPs, important details of the implementation have been left out. In this thesis, essential parts of the implementation have been thoroughly explained.

The results show how the different VEP methods compare using the low-cost EEG acquisition device, Emotiv EPOC. This can give an indication of which methods should be focused upon for further research and development, when using a headset of this kind. Method M2, when using a window length of three seconds and a threshold of 0.2, reached an average ITR of 32.92 bits/min. This is the highest ITR reported to date in a VEP-based BCI using Emotiv EPOC in an online setting. In

addition, it has been found that c-VEP methods have issues with EEG recordings using the standard DLLs that come with Emotiv EPOC.

6.4 Future Work

This section details what can be done to further the work presented in this thesis. The main issues and concerns are addressed, with suggestions for how they can be solved.

6.4.1 Emokit

There were two main issues encountered during the implementation and testing of the different VEP techniques. The first of these is the requirement from Emotiv EPOC for good sensor connectivity for the reference sensor. This requirement made half the test subjects unable to use Emotiv EPOC for VEP-based methods. The second issue is the lack of accurate timing information when implementing c-VEP. Both of these issues have roots in the DLLs that come with the SDK for Emotiv EPOC. Among other things, these DLLs give developers access to the EEG data. The EEG data is preprocessed, by applying Notch filters and subtracting the reference sensor from all other sensors, before the data is available to the developer. The data is also given in blocks of minimum four samples. These issues are explained in more detail in Section 4.3.4 and Section 4.2.

There is an open source Python module called Emokit[1]. It is a project where Emotiv EPOC has been reverse engineered so that the DLLs are no longer needed, and gives developers access to the raw, unfiltered EEG data. Using Emokit would solve both the aforementioned problems. First, the developer would be able to choose if and which sensor to use as the reference sensor. This would allow more persons to use VEP-based BCIs with Emotiv EPOC by using a reference sensor that already has a good fit. It could also lead to greater performance of the methods implemented in this thesis. The c-VEP method presented by Bin et al. [2] selected the optimal reference sensor for each subject as the sensor that, when subtracted from the EEG data, maximized training accuracy. This method could be used for improving classification accuracy for the SSVEP methods that were implemented in this thesis. The second issue regarding accurate timing information would also be solved. Since Emokit does not perform any preprocessing on the data, and sends the EEG data when it is available instead of in blocks of four, the developer would know with more certainty when the sample was recorded.

6.4.2 Further testing

It is clear that the results presented here do not support the findings by Cao et al. [5] that white is the best color for visual stimulation. The results suggest that the best color depends on the individual. As mentioned in Section 6.2, these results

[1] `https://github.com/openyou/emokit`

are based on a small sample size. The research could be furthered by performing more extensive color testing on a larger sample size to reach a conclusion on the subject.

More extensive testing should also be done for finding the optimal thresholds and window lengths for Emotiv EPOC with the different methods. Parameters can be tweaked to optimize accuracy and ITRs, as well as reducing detection times. More data from a larger quantity of test subjects is needed to accomplish this task.

Appendices

Appendix A

Structured literature review

This chapter gives a detailed description of how the SLR was conducted. Section A.1 states the three problems Pr1, Pr2 and Pr3 that the SLR is built upon, and defines the research questions that were asked to solve the problems. The search process is described in Section A.2, including the sources used in the SLR, the search words used and what the exclusion criteria are. Next, Section A.3 shows the study selection process that was applied on the search results to obtain a set of relevant articles. Section A.4 concludes the chapter with a description of how data collection and data analysis was performed on the set of articles.

A.1 Problems and Research Questions

The goal of this structured literature review (SLR) is to find information regarding the three problems presented in Section 2.4. These problems are

Pr1 Implementing techniques for VEP-based BCIs

Pr2 Implementing a VEP-based BCI with Emotiv EPOC

Pr3 Implementing a stimulus system for VEP-based BCIs

For each of these problems, the review aims to find what the existing solutions are, how they compare to each other, what the strength of the evidence for each solution is and what implications these solutions have. This can be rewritten into four research questions for each of the problems and are summarized in Table A.1.

A.2 Search Process

The search process consisted of searching through a selection of digital libraries and journals. The sources searched through are shown in table A.2.

For each of the three problems described above, a set of search keywords were

Pr1 Implementing techniques for VEP-based BCIs

RQ1	What are the existing techniques for VEP-based BCIs?
RQ2	How do the different solutions, found by addressing RQ1, compare to each other with respect to ITR?
RQ3	What is the strength of evidence in support of the different techniques?
RQ4	What implications will these findings have when implementing a VEP-based BCI technique?

Pr2 Implementing a VEP-based BCI with Emotiv EPOC

RQ1	What are the existing solutions for VEP-based BCIs implemented on Emotiv EPOC?
RQ2	How do the different solutions, found by addressing RQ1, compare to each other with respect to ITR?
RQ3	What is the strength of evidence in support of the different solutions?
RQ4	What implications will these findings have when implementing a VEP-based BCI with Emotiv EPOC?

Pr3 Implementing a stimulus system for VEP-based BCIs

RQ1	What are the existing stimulus system solutions for VEP-based BCIs?
RQ2	How do the different solutions, found by addressing RQ1, compare to each other with respect to ITR?
RQ3	What is the strength of evidence in support of the different solutions?
RQ4	What implications will these findings have when implementing a VEP-based BCI stimulus system?

Table A.1: Pr1, Pr2, Pr3 with four research questions.

Digital library	**URL**
IEEE Xplore	`http://ieeexplore.ieee.org`
SpringerLink	`http://www.springerlink.com`
ISI Web of Knowledge	`http://www.isiknowledge.com`
ScienceDirect	`http://www.sciencedirect.com`
Journal of Neural Engineering	`http://iopscience.iop.org/`

Table A.2: List of sources used for the structured literature review.

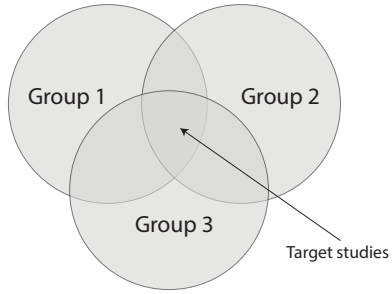

Figure A.1: Relevant studies. Adapted from Kofod-Petersen [24].

produced. Each keyword can have synonyms or words that provide a similar semantic meaning, and all of these terms are grouped together. The problems with their associated search keywords are shown in table A.3. The primary goal is to find the literature that is the intersection of the groups, see Figure A.1. A complete search involves searching for all permutations of keywords. **Pr1** has 27 permutations, while **Pr2** and **Pr3** have 18 permutations, bringing the total number of permutations to 63 for all three problems. The sources in table A.2 provide an *advanced search* option, and allows the use of Boolean keywords (AND ($\wedge$), OR ($\vee$)) when searching. When possible, the search was performed on the abstract, title and keywords of the article instead of a full text search. For **Pr1**, all permutations can be captured in the search string:

$$([G1,T1] \vee [G1,T2] \vee [G1,T3]) \wedge ([G2,T1] \vee [G2,T2] \vee [G2,T3]) \wedge ([G3,T1] \vee [G3,T2] \vee [G3,T3])$$

As table A.3 shows, all problems share two keywords, with the third keyword being unique to each distinct problem. To simplify the search process, the third column for each problem was combined into the search keywords shown in table A.4.

The combined search terms of table A.4 result in the following search string:

(BCI OR "brain computer interface" OR "brain computer interfaces") AND
(VEP OR "visual evoked potential" OR "visual evoked potentials") AND
((ITR OR "information transfer rate" OR "information transfer rates") OR
(emotiv OR epoc) OR ("visual stimulus" OR "visual stimulator"))

A prerequisite for using this search string is that the articles for each problem have the same inclusion and quality criteria. This prerequisite is fulfilled, and the criteria are discussed in detail in section A.3. The search string was used for all the sources shown in table A.2, and a total of 434 results were obtained. Before an article was passed on to the study selection process, it was necessary to exclude some of the articles. An article was excluded if it fulfilled at least one of three criteria:

- It was a duplicate from the same search engine

Pr1 Implementing techniques for VEP-based BCIs

	Group 1	**Group 2**	**Group 3**
Term 1	visual evoked potential	brain computer interface	information transfer rate
Term 2	visual evoked potentials	brain computer interfaces	information transfer rates
Term 3	VEP	BCI	ITR

Pr2 Implementing a VEP-based BCI with Emotiv EPOC

	Group 1	**Group 2**	**Group 3**
Term 1	visual evoked potential	brain computer interface	emotiv
Term 2	visual evoked potentials	brain computer interfaces	epoc
Term 3	VEP	BCI	

Pr3 Implementing a stimulus system for VEP-based BCIs

	Group 1	**Group 2**	**Group 3**
Term 1	visual evoked potential	brain computer interface	visual stimulus
Term 2	visual evoked potentials	brain computer interfaces	visual stimulator
Term 3	VEP	BCI	

Table A.3: Problems and their associated search terms

	Group 1	**Group 2**	**Group 3**
Term 1	visual evoked potential	brain computer interface	information transfer rate
Term 2	visual evoked potentials	brain computer interfaces	information transfer rates
Term 3	VEP	BCI	ITR
Term 4			emotiv
Term 5			epoc
Term 6			visual stimulus
Term 7			visual stimulator

Table A.4: Combined search terms from table A.3

Digital library	Total	PIC	SIC	QC
IEEE Xplore	81	45	15	9
SpringerLink	180	32	6	1
ISI Web of Knowledge	109	23	12	7
ScienceDirect	14	5	2	1
Journal of Neural Engineering	50	8	0	0
Combined	**434**	**113**	**35**	**18**

Table A.5: The amount of papers meeting the three categories of criteria.

- The same study was published in different sources
- It was published before January 1st, 2009

The reason for discarding articles published before January 1st, 2009 is because of the nature of the research domain. The fields of VEPs and BCIs are changing rapidly and articles before this date are considered outdated.

A.3 Study Selection Process

After searching through the sources as described in Section A.2, a three stage filtering process was applied. The goal of the filtering process was to narrow down the results to a set of papers that were relevant to the problems and research questions described in Section A.1. The three stages were:

1. Abstract inclusion criteria screening
2. Full-text inclusion criteria screening
3. Full-text quality screening

In each stage several inclusion criteria and quality screening criteria were applied. The criteria are outlined in Table A.6, and a bigger overview of the whole study selection process is shown in Figure A.2. The set containing IC1 and IC2 is referred to as the *primary inclusion criteria (PIC)*, and the set containing IC3, IC4 and IC5 is referred to as the *secondary inclusion criteria (SIC)*. The set containing the quality criteria is referred to as *quality criteria (QC)*. Table A.5 shows the statistics about the amount of papers meeting the three categories of criteria from each of the sources.

A.3.1 Criteria rationale

The PIC (IC1 and IC2) ensure that the articles found are relevant, and that the system or technique was implemented and tested. The SIC (IC3 and IC4) are meant to ensure that the articles present results that can be compared to other

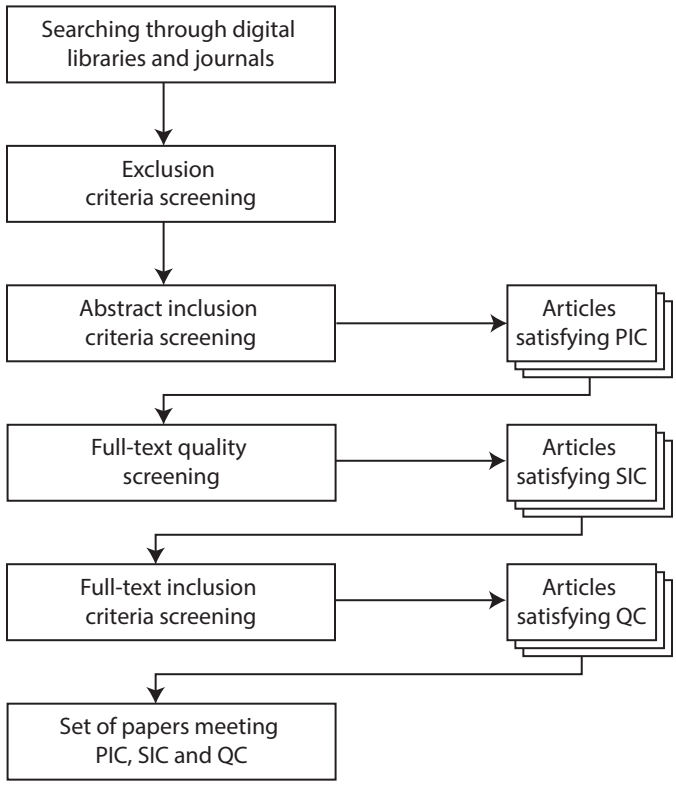

Figure A.2: The study selection process.

Criterion identifier	Criterion
IC1	The study's main concern is either Pr1, Pr2 or Pr3
IC2	The study presents empirical results
IC3	Presents results based on an online experiment
IC4	ITR is used as the performance metric for the results of the study
QC1	Is there a clear statement of the aim of the research?
QC2	Is the study put into context of other studies and research?
QC3	Is the study technique or algorithm thoroughly explained and reproducible?
QC4	Is the experimental procedure thoroughly explained and reproducible?
QC5	Does the test evidence support the findings presented?

Table A.6: The inclusion criteria and quality screening criteria applied in the study selection process.

methods by using the preferred performance metric for BCI systems (ITR), and that the experiments are performed in a relevant setting (online). Both of these topics are discussed in detail in Section 2.2.

The QC are used to evaluate the strength of the evidence presented in the articles. QC1 and QC2 are related to good research practice; only when both of these criteria are met can readers understand the assumptions that the research is based on, and what the research contributes to the field. QC3 and QC4 assesses to what degree the techniques and experimental setups are reproducible. It should be possible for readers of the article to reproduce the work presented in order to confirm the validity of the results. These criteria are particularly important for this thesis. QC5 assesses that all aspects of the results are analyzed, and that the conclusion of the article is backed by the findings presented.

A.3.2 Abstract screening

In this screening stage, the studies from the search process were filtered based on the title, abstract and keywords. The papers were accepted in this stage if the abstract indicated that the PIC were met (from Table A.6):

IC1 The study's main concern is either Pr1, Pr2 or Pr3

IC2 The study presents empirical results

The title and the abstract did not necessarily give sufficient information to conclude whether or not IC1 and IC2 were met. The policy used in such cases was to pass the paper on to the next filtering stage. The result of the abstract screening was that 321 studies were rejected, leaving 113 articles which met the PIC and were passed on to the full-text screening.

A.3.3 Full-Text Inclusion Criteria Screening

The abstract and title give limited information about an article. In this stage, the whole text was considered to determine if an article from the previous stage met the SIC (from Table A.6):

IC3 Presents results based on an online experiment

IC4 ITR is used as the performance metric for the results of the study

The papers that did not meet the SIC were discarded. The screening of the full text could also shed some light on whether or not the PIC were actually satisfied; the articles that did not fulfill the PIC were also discarded. After performing the full-text screening, the 113 articles were reduced to a set of 35 articles.

A.3.4 Full-Text Quality Assessment

The goal of the final step was to evaluate the strength of the evidence presented by the articles. This step thus evaluates RQ3 (listed in Table A.1). The articles that

passed this stage have strong evidence in support of their solutions and techniques. The articles were evaluated with respect to these criteria (from Table A.6):

QC1 Is there a clear statement of the aim of the research?

QC2 Is the study put into context of other studies and research?

QC3 Is the study technique or algorithm thoroughly explained and reproducible?

QC4 Is the experimental procedure thoroughly explained and reproducible?

QC5 Does the test evidence support the findings presented?

Each of the criteria were answered with yes (1 point), partially (0.5 points) or no (0 points). A threshold of 4.5 out of 5 points were set to pass this final stage. Table A.7 shows all the 35 articles, their individual score, and whether or not they passed the quality assessment stage. In total there were 18 articles that passed this stage. These articles are further analyzed in Chapter 3.

Table A.7: Quality assessment

Article ID	Authors	Title	Score	Pass
A1	Lopez-Gordo et al.	A high performance SSVEP-BCI without gazing	3.5	
A2	Cao et al.	A high rate online SSVEP based brain-computer interface speller	4.5	✓
A3	Cecotti	A Self-Paced and Calibration-Less SSVEP-Based Brain–Computer Interface Speller	5	✓
A4	Lee et al.	An SSVEP-Based BCI Using High Duty-Cycle Visual Flicker	4	
A5	Wang et al.	Developing stimulus presentation on mobile devices for a truly portable SSVEP-based BCI	4	
A6	Vilic et al.	DTU BCI speller– An SSVEP-based spelling system with dictionary support	4.5	✓
A7	Volosyak et al.	Evaluation of the Bremen SSVEP based BCI in real world conditions	4.5	✓
A8	Cao et al.	Flashing color on the performance of SSVEP-based brain-computer interfaces	4.5	✓
A9	Hwang et al.	Implementation of a mental spelling system based on steady-state visual evoked potential (SSVEP)	3	
A10	Liu et al.	Implementation of SSVEP Based BCI with Emotiv EPOC	5	✓
A11	Chang et al.	Independence of Amplitude-Frequency and Phase Calibrations in an SSVEP-Based BCI Using Stepping Delay Flickering Sequences	4	
A12	Han et al.	Modified pattern-reversal visual checkerboard stimuli with dual alternating frequencies for multi-class ssvep-based brain-computer interfaces	4	

Continued on next page

Table A.7 – continued from previous page

Article ID	Authors	Title	Score	Pass
A13	da Cruz et al.	Patterned visual stimuli for enhancement of SSVEP-based BCI performance	5	✓
A14	Chang et al.	Real-Time Control of an SSVEP-Actuated Remote-Controlled Car	4.5	✓
A15	Wang et al.	Visual stimulus design for high-rate SSVEP BCI	4.5	✓
A16	Zhang and Deng	An Automatic SSVEP Component Selection Measure for High-Performance Brain-Computer Interface	4	
A17	da Cruz et al.	An SSVEP-Based BCI with Adaptive Time-Window Length	5	✓
A18	Volosyak et al.	Impact of Frequency Selection on LCD Screens for SSVEP Based Brain-Computer Interfaces	4	
A19	Garcia-Molina and Zhu	Phase Detection of Visual Evoked Potentials Applied to Brain Computer Interfacing	4	
A20	Yan et al.	Right-and-left visual field stimulation A frequency and space mixed coding method for SSVEP based brain-computer interface	3.5	
A21	Segers et al.	Steady State Visual Evoked Potential (SSVEP) - Based Brain Spelling System with Synchronous and Asynchronous Typing Modes	4	
A22	Wang et al.	A cell-phone-based brain-computer interface for communication in daily life	4	
A23	Bin et al.	A high-speed BCI based on code modulation VEP	4.5	✓
A24	Hwang et al.	A new dual-frequency stimulation method to increase the number of visual stimuli for multi-class SSVEP-based brain-computer interface (BCI)	3.5	
A25	Bin et al.	An online multi-channel SSVEP-based brain-computer interface using a canonical correlation analysis method	5	✓

Continued on next page

Table A.7 – continued from previous page

Article ID	Authors	Title	Score	Pass
A26	Lee et al.	An SSVEP-Actuated Brain Computer Interface Using Phase-Tagged Flickering Sequences– A Cursor System	4	
A27	Diez et al.	Asynchronous BCI control using high-frequency SSVEP	5	✓
A28	Chen et al.	Brain–computer interface based on intermodulation frequency	4	
A29	Wu et al.	Frequency recognition in an SSVEP-based brain computer interface using empirical mode decomposition and refined generalized zero-crossing	4.5	✓
A30	Volosyak	SSVEP-based Bremen-BCI interface-boosting information transfer rates	5	✓
A31	Zhu et al.	Online BCI Implementation of High-Frequency Phase Modulated Visual Stimuli	4	
A32	Spüler et al.	Online Adaptation of a c-VEP Brain-Computer Interface(BCI) Based on Error-Related Potentials and Unsupervised Learning	5	✓
A33	Yeh et al.	Improvement of classification accuracy in a phase-tagged steady-state visual evoked potential-based brain computer interface using multiclass support vector machine	4.5	✓
A34	İşcan and Dokur	A novel steady-state visually evoked potential-based brain–computer interface design– Character Plotter	4	
A35	Chang et al.	An amplitude-modulated visual stimulation for reducing eye fatigue in SSVEP-based brain–computer interfaces	5	✓

A.4 Data Collection and Analysis

The study collection process resulted in 18 articles. A more detailed analysis was performed to collect information from the set of articles. The goal of this analysis was to gather information concerning RQ1 and RQ2 for the three problems Pr1, Pr2 and Pr3 (from Table A.1). These findings are presented in Chapter 3 along with a discussion regarding RQ4. RQ3 was handled in the full-text quality assessment (Section A.7).

The articles were read in detail and labeled with two types of labels. The first label marks which of the three problems, Pr1, Pr2, Pr3, the article addresses. The second label is the VEP paradigm the system implemented. The literature obtained either presented a SSVEP or a c-VEP system. Table A.8 lists each article identifier and what labels this article was given. In addition, the performance metric (ITR) of the systems are written in the last column.

Article ID	Problem	System	ITR
A32	Pr1	c-VEP	144.00
A23	Pr1	c-VEP	108.00
A30	Pr1	SSVEP	61.70
A2	Pr1	SSVEP	61.64
A17	Pr1	SSVEP	58.36
A25	Pr1	SSVEP	58.00
A33	Pr1	SSVEP	50.91
A14	Pr1	SSVEP	49.79
A35	Pr1	SSVEP	39.41
A3	Pr1	SSVEP	37.62
A29	Pr1	SSVEP	36.99
A7	Pr1	SSVEP	22.60
A6	Pr1	SSVEP	21.94
A27	Pr1	SSVEP	9.40-45.00
A10	Pr2	SSVEP	20.97
A15	Pr3	SSVEP	75.40
A13	Pr3	SSVEP	45.90
A8	Pr3	SSVEP	36.61

Table A.8: The articles listed with their labels and ITR. The table is sorted first on problem type, then by ITR.

Appendix B

Code

B.1 Canonical Correlation Analysis (CCA) Python Implementation

This section shows the Python implementation of MATLAB's *canoncorr* implementation to compute the canonical correlation between two sets of data, X and Y.

```
import numpy as np
from scipy.linalg import qr

def cca(X,Y):
  n,p1=X.shape if (len(X.shape) > 1) else (X.shape[0],1)
  p2=Y.shape[1] if (len(Y.shape) > 1) else 1

  X=X-np.tile(np.mean(X,0),(n,1))
  Y=Y-np.tile(np.mean(Y,0),(n,1))

  Q1,T11,perm1 = qr(X,pivoting=True,mode='economic')
  if len(T11.shape) > 1:
    val = np.spacing(np.absolute(T11[0][0]))*max(n,p1)
  else:
    val = np.spacing(np.absolute(T11[0]))*max(n,p1)
  rankX = sum(np.absolute(np.diag(T11).copy()) > val)

  Q2,T22,perm2 = qr(Y,pivoting=True,mode='economic')
  if len(T22.shape) > 1:
    val = np.spacing(np.absolute(T22[0][0]))*max(n,p2)
  else:
    val = np.spacing(np.absolute(T22[0]))*max(n,p2)
  rankY = sum(np.absolute(np.diag(T22).copy()) > val)

  d = min(rankX,rankY)
  L,D,Mh = np.linalg.svd(np.dot(Q1.T,Q2),full_matrices=False)
  D = np.diagflat(D)
  r = np.minimum(np.maximum(np.diag(D[:,:3]).T, 0), 1)
  M = Mh.T
  A = np.linalg.lstsq(T11,L[:,:d])[0]* np.sqrt(n-1)
  B = np.linalg.lstsq(T22,M[:,:d])[0]* np.sqrt(n-1)

  A[perm1,:] = np.vstack((A,np.zeros((p1-rankX,d))))
  B[perm2,:] = np.vstack((B,np.zeros((p2-rankY,d))))

  U = np.dot(X,A)
  V = np.dot(Y,B)
  return A,B,r,U,V
```

Bibliography

[1] 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.

[2] 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.

[3] 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.

[4] Bin, G., Lin, Z., Gao, X., Hong, B., and Gao, S. (2008). The ssvep topographic scalp maps by canonical correlation analysis. In *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, pages 3759–3762. IEEE.

[5] Cao, T., Wan, F., Mak, P. U., Mak, P.-I., Vai, M. I., and Hu, Y. (2012). Flashing color on the performance of ssvep-based brain-computer interfaces. In *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*, pages 1819–1822. IEEE.

[6] Cao, T., Wang, X., Wang, B., Wong, C. M., Wan, F., Mak, P. U., Mak, P. I., and Vai, M. I. (2011). A high rate online ssvep based brain-computer interface speller. In *Neural Engineering (NER), 2011 5th International IEEE/EMBS Conference on*, pages 465–468. IEEE.

[7] Cecotti, H. (2010). A self-paced and calibration-less ssvep-based brain–computer interface speller. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 18(2):127–133.

[8] Chang, H.-C., Deng, H.-T., Lee, P.-L., Wu, C.-H., and Shyu, K.-K. (2010). Real-time control of an ssvep-actuated remote-controlled car. In *SICE Annual Conference 2010, Proceedings of*, pages 1884–1887. IEEE.

[9] Chang, H.-C., Lee, P.-L., Lo, M.-T., Lee, I.-H., Yeh, T.-K., and Chang, C.-Y. (2012). Independence of amplitude-frequency and phase calibrations in an ssvep-based bci using stepping delay flickering sequences. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 20(3):305–312.

[10] Chang, M. H., Baek, H. J., Lee, S. M., and Park, K. S. (2013). An amplitude-modulated visual stimulation for reducing eye fatigue in ssvep-based brain–computer interfaces. *Clinical Neurophysiology*.

[11] Chen, X., Chen, Z., Gao, S., and Gao, X. (2013). Brain–computer interface based on intermodulation frequency. *Journal of neural engineering*, 10(6):066009.

[12] Condon, J. J. and S. M. Ransom, N. R. A. O. (2010). `http://www.cv.nrao.edu/course/astr534/FourierTransforms.html`.

[13] da Cruz, J. N., Wong, C. M., Cao, T., and Wan, F. (2013a). 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.

[14] da Cruz, J. N., Wong, C. M., and Wan, F. (2013b). An ssvep-based bci with adaptive time-window length. In *Advances in Neural Networks–ISNN 2013*, pages 305–314. Springer.

[15] Diez, P. F., Mut, V. A., Perona, E. M. A., and Leber, E. L. (2011). Asynchronous bci control using high-frequency ssvep. *Journal of neuroengineering and rehabilitation*, 8(1):39.

[16] Duvinage, M., Castermans, T., Petieau, M., Hoellinger, T., Cheron, G., and Dutoit, T. (2013). Performance of the emotiv epoc headset for p300-based applications. *Biomedical engineering online*, 12(1):56.

[17] Friman, O., Volosyak, I., and Graser, A. (2007). Multiple channel detection of steady-state visual evoked potentials for brain-computer interfaces. *Biomedical Engineering, IEEE Transactions on*, 54(4):742–750.

[18] Garcia-Molina, G. and Zhu, D. (2013). Phase detection of visual evoked potentials applied to brain computer interfacing. In *Towards Practical Brain-Computer Interfaces*, pages 269–280. Springer.

[19] Han, C.-H., Hwang, H.-J., and Im, C.-H. (2013). Modified pattern-reversal visual checkerboard stimuli with dual alternating frequencies for multi-class ssvep-based brain-computer interfaces. In *Brain-Computer Interface (BCI), 2013 International Winter Workshop on*, pages 86–88. IEEE.

[20] Hwang, H.-J., Hwan Kim, D., Han, C.-H., and Im, C.-H. (2013a). A new dual-frequency stimulation method to increase the number of visual stimuli for multi-class ssvep-based brain–computer interface (bci). *brain research*, 1515:66–77.

[21] Hwang, H.-J., Lim, J.-H., Lee, J.-H., and Im, C.-H. (2013b). Implementation of a mental spelling system based on steady-state visual evoked potential (ssvep). In *Brain-Computer Interface (BCI), 2013 International Winter Workshop on*, pages 81–83. IEEE.

[22] Instruments, N. (2014). `http://zone.ni.com/reference/en-XX/help/372416B-01/svtconcepts/fft_funda/`.

[23] İşcan, Z. and Dokur, Z. (2013). A novel steady-state visually evoked potential-based brain–computer interface design: Character plotter. *Biomedical Signal Processing and Control*.

[24] Kofod-Petersen, A. (2012). How to do a structured literature review in computer science.

[25] Lee, P.-L., Hsieh, J.-C., Wu, C.-H., Shyu, K.-K., Chen, S.-S., Yeh, T.-C., and Wu, Y.-T. (2006). The brain computer interface using flash visual evoked potential and independent component analysis. *Annals of biomedical engineering*, 34(10):1641–1654.

[26] Lee, P.-L., Sie, J.-J., Liu, Y.-J., Wu, C.-H., Lee, M.-H., Shu, C.-H., Li, P.-H., Sun, C.-W., and Shyu, K.-K. (2010). An ssvep-actuated brain computer interface using phase-tagged flickering sequences: a cursor system. *Annals of biomedical engineering*, 38(7):2383–2397.

[27] Lee, P.-L., Yeh, C.-L., Cheng, J.-S., Yang, C.-Y., and Lan, G.-Y. (2011). An ssvep-based bci using high duty-cycle visual flicker. *Biomedical Engineering, IEEE Transactions on*, 58(12):3350–3359.

[28] 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.

[29] Liu, Y., Jiang, X., Cao, T., Wan, F., Mak, P. U., Mak, P.-I., and Vai, M. I. (2012). Implementation of ssvep based bci with emotiv epoc. In *Virtual Environments Human-Computer Interfaces and Measurement Systems (VECIMS), 2012 IEEE International Conference on*, pages 34–37. IEEE.

[30] Lopez-Gordo, M. A., Pelayo, F., and Prieto, A. (2010). A high performance ssvep-bci without gazing. In *Neural Networks (IJCNN), The 2010 International Joint Conference on*, pages 1–5.

[31] 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.

[32] McFarland, D. J. and Wolpaw, J. R. (2011). Brain-computer interfaces for communication and control. *Communications of the ACM*, 54(5):60–66.

[33] Nunez, P. L. and Srinivasan, R. (2007). Electroencephalogram. *Scholarpedia*, 2(2):1348. revision 91218.

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

[35] Segers, H., Combaz, A., Manyakov, N. V., Chumerin, N., Vanderperren, K., Van Huffel, S., and Van Hulle, M. (2011). Steady state visual evoked potential (ssvep)-based brain spelling system with synchronous and asynchronous typing modes. In *15th Nordic-Baltic Conference on Biomedical Engineering and Medical Physics (NBC 2011)*, pages 164–167. Springer.

[36] Society, A. C. N. et al. (2006). Guideline 5: Guidelines for standard electrode position nomenclature. *American journal of electroneurodiagnostic technology*, 46(3):222.

[37] Speier, W., Arnold, C., and Pouratian, N. (2013). Evaluating true bci communication rate through mutual information and language models. *PloS one*, 8(10):e78432.

[38] Spüler, M., Rosenstiel, W., and Bogdan, M. (2012). Online adaptation of a c-vep brain-computer interface (bci) based on error-related potentials and unsupervised learning. *PloS one*, 7(12):e51077.

[39] Stytsenko, K., Jablonskis, E., and Prahm, C. (2011). Evaluation of consumer eeg device emotiv epoc. In *MEi: CogSci Conference 2011, Ljubljana*.

[40] Townsend, G., LaPallo, B., Boulay, C., Krusienski, D., Frye, G., Hauser, C., Schwartz, N., Vaughan, T., Wolpaw, J., and Sellers, E. (2010). A novel p300-based brain–computer interface stimulus presentation paradigm: moving beyond rows and columns. *Clinical Neurophysiology*, 121(7):1109–1120.

[41] van Erp, J., Lotte, F., and Tangermann, M. (2012). Brain-computer interfaces: Beyond medical applications. *Computer*, 45(4):26–34.

[42] Vilic, A., Kjaer, T. W., Thomsen, C. E., Puthusserypady, S., and Sorensen, H. B. (2013). Dtu bci speller: An ssvep-based spelling system with dictionary support. In *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE*, pages 2212–2215. IEEE.

[43] Volosyak, I. (2011). Ssvep-based bremen–bci interface—boosting information transfer rates. *Journal of neural engineering*, 8(3):036020.

[44] Volosyak, I., Cecotti, H., and Gräser, A. (2009a). Impact of frequency selection on lcd screens for ssvep based brain-computer interfaces. In *Bio-Inspired Systems: Computational and Ambient Intelligence*, pages 706–713. Springer.

[45] Volosyak, I., Cecotti, H., Valbuena, D., and Graser, A. (2009b). Evaluation of the bremen ssvep based bci in real world conditions. In *Rehabilitation Robotics, 2009. ICORR 2009. IEEE International Conference on*, pages 322–331. IEEE.

[46] Wang, Y., Wang, Y.-T., and Jung, T.-P. (2010). Visual stimulus design for high-rate ssvep bci. *Electronics Letters*, 46(15):1057–1058.

[47] Wang, Y.-T., Wang, Y., Cheng, C.-K., and Jung, T.-P. (2013). Developing stimulus presentation on mobile devices for a truly portable ssvep-based bci. In *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE*, pages 5271–5274. IEEE.

[48] Wang, Y.-T., Wang, Y., and Jung, T.-P. (2011). A cell-phone-based brain–computer interface for communication in daily life. *Journal of neural engineering*, 8(2):025018.

[49] Wolpaw, J. R., Ramoser, H., McFarland, D. J., and Pfurtscheller, G. (1998). Eeg-based communication: improved accuracy by response verification. *Rehabilitation Engineering, IEEE Transactions on*, 6(3):326–333.

[50] Wu, C.-H., Chang, H.-C., Lee, P.-L., Li, K.-S., Sie, J.-J., Sun, C.-W., Yang, C.-Y., Li, P.-H., Deng, H.-T., and Shyu, K.-K. (2011). Frequency recognition in an ssvep-based brain computer interface using empirical mode decomposition and refined generalized zero-crossing. *Journal of neuroscience methods*, 196(1):170–181.

[51] Yan, Z., Gao, X., and Gao, S. (2011). Right-and-left visual field stimulation: A frequency and space mixed coding method for ssvep based brain-computer interface. *Science China Information Sciences*, 54(12):2492–2498.

[52] 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.

[53] Yuan, P., Gao, X., Allison, B., Wang, Y., Bin, G., and Gao, S. (2013). A study of the existing problems of estimating the information transfer rate in online brain–computer interfaces. *Journal of neural engineering*, 10(2):026014.

[54] Zbilut, J. P. and Marwan, N. (2008). The wiener–khinchin theorem and recurrence quantification. *Physics Letters A*, 372(44):6622–6626.

[55] Zhang, Z. and Deng, Z. (2014). An automatic ssvep component selection measure for high-performance brain-computer interface. In *Foundations and Practical Applications of Cognitive Systems and Information Processing*, pages 93–108. Springer.

[56] 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.

[57] Zhu, D., Garcia-Molina, G., Mihajlović, V., and Aarts, R. M. (2011). Online bci implementation of high-frequency phase modulated visual stimuli. In *Universal Access in Human-Computer Interaction. Users Diversity*, pages 645–654. Springer.

Abbreviations

AM amplitude-modulated.

BCI brain-computer interface.

CCA canonical correlation analysis.

CPM characters per minute.

CPU central processing unit.

c-VEP code-modulated VEP.

DC direct current.

DFT discrete Fourier transform.

DLL dynamic-link library.

DSB double-sideband suppressed carrier.

EEG electroencephalogram.

EP evoked potential.

ERD event-related desynchronization.

ERP event-related potential.

ErrP error-related potential.

FFT fast Fourier transform.

FIFO First in, First Out.

FPGA field-programmable gate array.

FVEP flash VEP.

IIR infinite impulse response.

ITR information transfer rate.

LCD liquid crystal display.

LDA linear discriminant analysis.

LED light-emitting diode.

MEC minimum energy combination.

MEG magnetoencephalography.

OCSVM one class support vector machine.

PBR practical bit rate.

PIC primary inclusion criteria.

PSD power spectral density.

PSDA power spectral density analysis.

QC quality criteria.

SDK software development kit.

SIC secondary inclusion criteria.

SLR structured literature review.

SMR sensorimotor rhythm.

SNR signal-to-noise ratio.

SSVEP steady state visually evoked potential.

SVM support vector machine.

t-VEP time-modulated VEP.

VEP visual evoked potential.