

**Towards a home-use BCI:
fast asynchronous control and
robust non-control state detection**

DISSERTATION

der Mathematisch-Naturwissenschaftlichen Fakultät
der Eberhard Karls Universität Tübingen
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)

vorgelegt von
SEBASTIAN NAGEL
aus Gelnhausen

Tübingen
2019

Gedruckt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät
der Eberhard Karls Universität Tübingen.

Tag der mündlichen Qualifikation: 29. November 2019

Dekan:

Prof. Dr. Wolfgang Rosenstiel

1. Berichterstatter:

Prof. Dr. Volker Franz

2. Berichterstatter:

Prof. Dr. Moritz Grosse-Wentrup

Abstract

Brain-Computer Interfaces (BCIs) enable users to control a computer by using pure brain activity. Their main purpose is to restore several functionalities of motor disabled people, for example, to restore the communication ability.

Recent BCIs based on visual evoked potentials (VEPs), which are brain responses to visual stimuli, have shown to achieve high-speed communication. However, BCIs have not really found their way out of the lab yet. This is mainly because all recent high-speed BCIs are based on synchronous control, which means commands can only be executed in time slots controlled by the BCI. Therefore, the user is not able to select a command at his own convenience, which poses a problem in real-world applications. Furthermore, all those BCIs are based on stimulation paradigms which restrict the number of possible commands. To be suitable for real-world applications, a BCI should be asynchronous, or also called self-paced, and must be able to identify the user's intent to control the system or not. Although there some asynchronous BCI approaches, none of them achieved suitable real-world performances.

In this thesis, the first asynchronous high-speed BCI is proposed, which allows using a virtually unlimited number of commands. Furthermore, it achieved a nearly perfect distinction between intentional control (IC) and non-control (NC), which means commands are only executed if the user intends to. This was achieved by a completely different approach, compared to recent methods. Instead of using a classifier trained on specific stimulation patterns, the presented approach is based on a general model that predicts arbitrary stimulation patterns. The approach was evaluated with a "traditional" as well as a deep machine learning method.

The resultant asynchronous BCI outperforms recent methods by a multi-fold in multiple disciplines and is an essential step for moving BCI applications out of the lab and into real life. With further optimization, discussed in this thesis, it could evolve to the very first end-user suitable BCI, as it is effective (high accuracy), efficient (fast classifications), ease of use, and allows to perform as many different tasks as desired.

Zusammenfassung

Eine Hirn-Computer Schnittstelle (engl. *Brain-Computer Interface*, BCI) erlaubt einem Nutzer einen Computer nur mittels Gehirn-Aktivität zu steuern. Der Hauptanwendungszweck ist die Wiederherstellung verschiedener Funktionen von motorisch eingeschränkten Menschen, zum Beispiel, die Wiederherstellung der Kommunikationsfähigkeit.

Bisherige BCIs die auf visuell evozierten Potentialen (VEPs) basieren, erlauben bereits hohe Kommunikationsgeschwindigkeiten. VEPs sind Reaktionen, die im Gehirn durch visuelle Stimulation hervorgerufen werden. Allerdings werden bisherige BCIs hauptsächlich in der Forschung verwendet und sind nicht für reale Anwendungszwecke geeignet. Grund dafür ist, dass sie auf dem synchronen Steuerungsprinzip beruhen, dies bedeutet, dass Aktionen nur in vorgegebenen Zeitslots ausgeführt werden können. Dies bedeutet wiederum, dass der Nutzer keine Aktionen nach seinem Belieben ausführen kann, was für reale Anwendungszwecke ein Problem darstellt. Um dieses Problem zu lösen, müssen BCIs die Intention des Nutzers, das System zu steuern oder nicht, erkennen. Solche BCIs werden asynchron oder selbstbestimmt genannt. Bisherige asynchrone BCIs zeigen allerdings keine ausreichende Genauigkeit bei der Erkennung der Intention und haben zudem eine deutlich reduzierte Kommunikationsgeschwindigkeit im Vergleich zu synchronen BCIs.

In dieser Doktorarbeit wird das erste asynchrone BCI vorgestellt, welches sowohl eine annäherungsweise perfekte Erkennung der Intention des Nutzers als auch eine ähnliche Kommunikationsgeschwindigkeit wie synchrone BCIs erzielt. Dies wurde durch die Entwicklung eines allgemeinen Modells für die Vorhersage von sensorischen Reizen erzielt. Dadurch können beliebige visuelle Stimulationsmuster basierend auf den gemessenen VEPs vorhergesagt werden. Das Modell wurde sowohl mit einem „traditionellen“ maschinellen Lernverfahren als auch mit einer *deep-learning* Methode implementiert und evaluiert.

Das resultierende asynchrone BCI übertrifft bisherige Methoden in mehreren Disziplinen um ein Vielfaches und ist ein wesentlicher Schritt, um BCI-Anwendungen aus dem Labor in die Praxis zu bringen. Durch weitere Optimierungen, die in dieser Arbeit diskutiert werden, könnte es sich zum allerersten geeigneten BCI für Endanwender entwickeln, da es effektiv (hohe Genauigkeit), effizient (schnelle Klassifizierungen), und einfach zu bedienen ist. Ein weiteres Alleinstellungsmerkmal ist, dass das entwickelte BCI für beliebige Szenarien verwendet werden kann, da es annähernd unendlich viele gleichzeitige Aktionsfelder erlaubt.

Acknowledgements

First, I would like to thank Prof. Dr. Wolfgang Rosenstiel for giving me the opportunity to write this thesis at the department of computer engineering and for funding this project. I would also like to thank Prof. Dr. Volker Franz and Prof. Dr. Moritz Grosse-Wentrup who agreed to be my supervisors at short notice.

A special thanks goes to Dr. Martin Spüler for his supervision during the period of this work and for his always-available support. Furthermore, I would like to thank all my colleagues, Tanja Krumpe, Christian Niethammer, Jörg Peter, and Katrin Sippel for giving constructive criticism. Especially Tanja Krumpe was always available as a participant for many experimental test phases.

I would also like to thank all my friends and especially my girlfriend, Kristina Heller, who have always helped me to clear my mind in stressful phases. Last but not least, I want to thank both my parents Harald Nagel und Ingrid Nagel, who always supported me. Moreover, I want to dedicate this work to my father, who fortunately convinced me to do my doctorate, but unfortunately he died during the period of this work.

Contents

1	Introduction	1
2	Fundamentals	3
2.1	Brain-Computer Interface	3
2.1.1	Active BCI	3
2.1.2	Passive BCI	4
2.1.3	Synchronous control	4
2.1.4	Asynchronous control	5
2.2	Recording techniques	5
2.2.1	Electroencephalography	6
2.3	Event-related potentials	7
2.3.1	Visual evoked potentials	7
2.4	Paradigms for BCI control	9
2.4.1	Steady-state VEP	9
2.4.2	Code modulated VEP	10
2.4.3	P300	10
2.5	Machine learning	11
2.5.1	Ridge regression	12
2.5.2	Convolutional neural network	13
2.5.2.1	Operations	14
2.5.2.2	Hyperparameters	16
2.6	Performance evaluation	17
2.6.1	Accuracy	17

2.6.2	Information transfer rate	17
2.6.3	Pearson Correlation	18
3	State of the art	19
3.1	Synchronous Brain-Computer-Interfaces	19
3.1.1	P300	19
3.1.2	cVEP	20
3.1.3	SSVEP	21
3.2	Asynchronous BCI / Non-control detection	22
3.3	Home-use Brain-Computer-Interfaces	23
3.4	Brain response prediction to visual stimuli	24
4	Visual stimulation framework for high synchronicity	27
4.1	General description	28
4.2	Synchronization with an EEG amplifier	28
4.3	Stimulation paradigms	29
4.3.1	cVEP	29
4.3.2	SSVEP	29
4.3.3	Random	30
4.4	Layouts	30
4.4.1	Matrix keyboard	30
4.4.2	QWERTZ keyboard	30
5	Monitor raster latency and its effect	33
5.1	Methods	34
5.1.1	Measuring raster latencies	34
5.1.2	Analysis of SSVEP data	34
5.1.2.1	Setup	34
5.1.2.2	Experimental design	35
5.1.2.3	Processing	35
5.1.3	Analysis of P300 data	35
5.1.3.1	Setup	35

5.1.3.2	Processing	36
5.1.4	Analysis of cVEP BCI data	36
5.1.4.1	Setup	36
5.1.4.2	Experimental design	36
5.1.4.3	Processing	36
5.1.4.4	Correcting raster latencies	37
5.1.4.5	Performance evaluation	37
5.2	Results	37
5.2.1	Measuring raster latencies	37
5.2.2	Effects on VEP generation	38
5.2.3	Effects on P300	40
5.2.4	Effects on BCI control performance	40
5.3	Discussion	41
5.3.1	Effects of raster latencies	41
5.3.2	Addressing raster latencies	43
6	EEG2Code: predicting stimulation patterns	45
6.1	Methods	46
6.1.1	Training	46
6.1.2	Prediction	47
6.1.3	Random modulation patterns	48
6.1.4	Hardware & Software	48
6.1.5	Data acquisition	49
6.1.6	Preprocessing	49
6.1.6.1	Frequency filter	49
6.1.6.2	Correcting raster latencies	50
6.1.6.3	Spatial filter	50
6.1.7	Performance evaluation	50
6.2	Results	51
6.3	Discussion	51
7	Code2EEG: predicting brain responses to visual stimuli	53

7.1	Methods	54
7.1.1	Training	54
7.1.2	Prediction	54
7.1.3	Hardware & Software	54
7.1.4	Data acquisition	56
7.1.5	Performance evaluation	56
7.2	Results	57
7.3	Discussion	59
8	Optimization of stimulation patterns	61
8.1	Methods	62
8.1.1	Analyzing the m-sequence	62
8.1.2	Effects of the number of bit-changes	62
8.1.3	Generate optimized stimulation patterns	62
8.1.4	Hardware & Software	63
8.1.5	Data acquisition	64
8.1.6	Performance evaluation	64
8.2	Results	64
8.3	Discussion	65
9	Synchronous BCI control	69
9.1	Methods	70
9.1.1	Target classification	70
9.1.1.1	Hamming distance	70
9.1.1.2	Euclidean distance	70
9.1.1.3	Correlation	71
9.1.2	Hardware & Software	71
9.1.3	Data acquisition	72
9.1.4	Comparison to a cVEP BCI speller	72
9.1.5	Performance evaluation	73
9.2	Results	73
9.2.1	Comparison of the different target classification methods . . .	73

9.2.2	Comparison of EEG2Code and Code2EEG	74
9.2.3	Comparison of random and optimized stimulation patterns	74
9.2.4	Comparison to a cVEP BCI speller	75
9.3	Discussion	76
10	Increasing the number of targets and varying trial durations	79
10.1	Methods	80
10.1.1	Varying trial duration and number of targets	80
10.1.2	Hardware & Software	80
10.1.3	Data acquisition	81
10.1.4	Performance evaluation	81
10.2	Results	82
10.3	Discussion	85
11	Asynchronous BCI control	87
11.1	Methods	88
11.1.1	Asynchronous BCI control	88
11.1.2	Threshold determination	88
11.1.3	Hardware & Software	90
11.1.4	Data acquisition	91
11.1.5	Performance evaluation	91
11.2	Results	92
11.2.1	Online lexicographic spelling performance	92
11.2.2	Online non-control detection performance	93
11.2.3	Online case-sensitive copy-spelling performance	93
11.2.4	Offline threshold optimization	93
11.3	Discussion	95
12	Effects of varying synchronization latencies	99
12.1	Methods	99
12.1.1	Simulate synchronization latencies	99
12.1.2	Hardware & Software	100

12.1.3	Data acquisition	101
12.1.4	Performance evaluation	101
12.2	Results	101
12.3	Discussion	102
13	Improve performance using deep-learning	105
13.1	Methods	106
13.1.1	Convolutional EEG2Code model	106
13.1.2	Hardware & Software	108
13.1.3	Data acquisition	108
13.1.4	Performance evaluation	109
13.2	Results	109
13.2.1	Offline analysis: Simulated online experiment	109
13.2.2	Online experiment	110
13.3	Discussion	111
14	Reduced training data and cross-subject performance	115
14.1	Methods	116
14.1.1	Reduce training data	116
14.1.2	Cross-subject	116
14.1.3	Hardware & Software	116
14.1.4	Data acquisition	117
14.1.5	Performance evaluation	117
14.2	Results	118
14.3	Discussion	120
15	Peripheral perception of different colors	121
15.1	Methods	122
15.1.1	Analyze peripheral perception	122
15.1.2	Stimulation pattern	122
15.1.3	Color selection	123
15.1.4	Analysis	124

CONTENTS

xi

15.1.5 Hardware & Software	125
15.1.6 Data acquisition	125
15.1.7 Preprocessing	126
15.2 Results	126
15.3 Discussion	127
16 Summary	131
17 Discussion	133
List of Abbreviations	137
Bibliography	139

Chapter 1

Introduction

In the 1960s, Paulo Freire stated that "only through communication can human life hold meaning" [Fre70]. Communication is important for interpersonal relationships and can be verbal or non-verbal, whereby the former is most important as it allows to express yourself precisely. Unfortunately, several diseases can lead to a loss of speech, or they can even lead to a complete loss of communication ability. For example, amyotrophic lateral sclerosis (ALS) causes a loss of voluntary muscle control, whereby mental health is generally unaffected, which means patients are perceiving what is happening. Until today there is no cure and studies have reported that between 75% and 95% of ALS patients will lose the ability of speech [SWST81,BFN11].

Being trapped in one's own body and especially no longer being able to communicate with relatives must be awful. Over the last years, Brain-Computer Interfaces (BCIs) have evolved as a helpful method to restore several functionalities of motor disabled people. BCIs enable users to control a computer by using brain activity, which allows, for example, to spell using a virtual keyboard. Recent BCI spellers are mainly based on event-related potentials (ERPs) or visual evoked potentials (VEPs). The latter are brain responses to visual stimuli and the idea to use it for communication has been proposed by Sutter, who stated that "the electrical scalp response to a modulated target is largest if the target is located within the central 1° of the visual field" and that "this makes it possible to construct a gaze-controlled keyboard" [Sut84].

Although recent BCI spellers [SRB12b,CWN⁺15] show high communication speed, they are based on synchronous control, which means that commands are executed in a certain time interval controlled by the BCI. However, those BCIs are not suitable for real-world applications as they cannot differentiate between intentional control and non-control state and will give a random output if a user is taking a break to think or does not want to control the BCI for other reasons.

Furthermore, a user would intend to use the BCI at home and not only in a lab. But, as recent BCIs are not really suitable for home-use, just a few were evaluated at

home. In order that a BCI system can be used at home, it must fulfill some criteria, whereby the most important are effectiveness, reliability, efficiency, functionality, and mobile usage [MVD⁺15, HBKK15]. This means, the user wants to perform as many as possible different tasks with the best possible accuracy, and those tasks should be performed as fast as possible. Furthermore, the BCI should work reliably for several years without malfunctioning and should optimally be usable on-the-go.

The goal of this work is to develop a BCI which addresses and improves the above-mentioned criteria. In the following, an outline of this thesis will be given.

In chapter 2 the required fundamentals are explained, including BCIs, brain signal recording techniques, and machine learning methods. In chapter 3 the history, as well as the current state-of-the-art of several BCI approaches, are given. In chapter 4 a stimulation framework is proposed which allows to easily create different stimulation layouts and ensures high synchronicity between the stimuli and the measured EEG. The effects of an inaccurate synchronization to the performance of VEP BCIs will be shown in chapter 5.

In chapter 6 an entirely new and unique stimulation paradigm is proposed, based on fully random stimulation patterns. Based on this, a method (EEG2Code) is introduced which is able to predict arbitrary stimulation patterns based on the measured VEP responses. In chapter 7 the backward approach (Code2EEG) is explained which allows predicting the brain response to arbitrary stimulation patterns. In chapter 8 an optimized set of random stimulation patterns is introduced which increases the performance compared to fully random stimulation patterns.

In chapter 9 it is proven that the EEG2Code, as well as the Code2EEG approach, can be used for synchronous BCI control. In chapter 10 it will be shown that both approaches can be used with a nearly unlimited number of targets. In chapter 11 the EEG2Code approach was extended to allow asynchronous control and non-control state detection.

In chapter 12 the mobile usage, which generally causes synchronization latencies, was addressed and it will be shown that the approach can handle such latencies. In chapter 13 the EEG2Code model was improved by using a deep-learning approach. As collecting training data takes time and is inconvenient for the user, the performance of the EEG2Code model using fewer train data was analyzed in chapter 14. In chapter 15 it will be shown that different stimuli colors lead to a different peripheral perception of neighbored targets. Finally, in chapters 16 and 17 the results of this thesis will be summarized and discussed.

Chapter 2

Fundamentals

2.1 Brain-Computer Interface

In 1973, the term "Brain-Computer Interface" (BCI) was coined by Jacques J. Vidal. In the context of a BCI project, he considered whether the "observable electrical brain signals can be put to work as carriers of information in man-computer communication or for the purpose of controlling such external apparatus as prosthetic devices or spaceships" [Vid73]. A few years later, in 1977, Vidal described the first real-time BCI application [Vid77], which was based on "visual event-related potentials", actually called visual evoked potential (VEP, see section 2.4).

Today, their main purpose is to restore several functionalities of motor disabled people, for example, patients who suffered a stroke or have amyotrophic lateral sclerosis (ALS). In general, a BCI can be understood as a communication pathway between the brain and an external device and they can be divided into two types: *passive* BCI and *active* BCI. Furthermore, a BCI can operate in two different control modes: *synchronous* or *asynchronous*. All those terms are important for the scope of this work and will be explained in the following subsections.

2.1.1 Active BCI

As mentioned, the main purpose of a BCI is to restore several functionalities of motor disabled people. For example, to restore communication [BGW⁺11, SRB12a, CWN⁺15], to control prostheses [HSF⁺06, HBJ⁺12, CWD⁺13], or to control a wheelchair [LFMP⁺07, GNL⁺08, LPWY13] in order to give patients back parts of their independence, and therefore, to improve the quality of life.

Those types of BCIs, are called *active* BCIs or *explicit* BCIs, as they enable users to actively perform commands to control the corresponding environment. Zander *et al.* defined an active BCI as "a BCI which derives its outputs from brain activity which

is directly consciously controlled by the user, independently from external events, for controlling an application" [ZKWR08].

2.1.2 Passive BCI

Contrary to an active BCI, a BCI can also be used in a passive manner. For example, for workload detection [KSR⁺18] or emotion detection [MHH17, AC16]. Those types of BCIs, are called *passive* BCIs or *implicit* BCIs, as the user is not able to explicitly perform commands.

In general, an implicit interaction can be defined as "an action performed by the user that is not primarily aimed to interact with a computerized system but which such a system understands as input" [Sch00]. Zander *et al.* defined a passive BCI as "a BCI which derives its outputs from arbitrary brain activity without the purpose of voluntary control, for enriching a human-computer interaction with implicit information" [ZKWR08].

It is worth to mention that an active BCI can be based on explicit information, like motor imagery, or on implicit information, like event-related potentials or visual evoked potentials (see section 2.4).

Furthermore, there are combinations of both. For example, Spüler *et al.* developed a BCI that allows a user to actively perform commands, whereby the underlying model is adaptively improved using implicit information, in that case, error-related potentials, which are potentials that occur shortly after the user recognizes an error [SRB12b].

2.1.3 Synchronous control

Whether explicit and implicit, the control mode of a BCI is a fundamental property. Most of recent BCIs make use of a synchronous control mode, which means that commands are executed in a certain time interval controlled by the BCI.

For example, the fastest synchronous active BCI to-date by Chen *et al.* [CWN⁺15] defines that the stimulation time lasts for 0.5 s with an additional 0.5 s of inter-stimulation time. This means, the user has to fixate the desired command in the specified time slot, otherwise the BCI will perform a random classification. Furthermore, due to the used method, there are only 280 ms between command classification and the start of the next stimulation, which is only suitable for experienced users.

Obviously, the advantage of synchronous control is, that the BCI does not have to distinguish if the user wants to control the BCI or not. Contrary, the user must know in advance exactly which commands he wants to perform. Furthermore, if

the user no longer wants to execute commands, the BCI must be stopped and then reactivated if the user wants to continue. For motor disabled people this is impossible and requires a third party. Therefore, synchronous BCIs are not suitable for real-world applications.

2.1.4 Asynchronous control

A practical BCI should be asynchronous, or also-called self-paced. The major task is to identify the user's intent to control the system, which is called the "Midas Touch" problem [Moo03]. The BCI has to distinguish efficiently between the intentional control (IC) state and the non-control (NC) state.

It must be noted that the term "asynchronous" is not uniformly defined. On the one hand, it is used for early-stopping methods disregarding the NC state detection, which means that the term is used to express that the duration of a trial is not strictly defined by the BCI, but implicitly driven by the user. On the other hand, it is used as explained above, for BCIs which can distinguish between IC and NC state. The method proposed in this work allows both early-stopping and classification of the user's intent to control the system, which is why it is called asynchronous.

It also should be mentioned that no unified criteria exist for the evaluation of NC state detection, which means it is not defined what a user should/could do during the NC state.

2.2 Recording techniques

As the name implies, a BCI is based on brain activity, which must be recorded in any manner. There are many recording techniques, which exploit different underlying physiological processes. For example, there is an increased oxygen consumption and an increased blood flow in active brain areas, called hemodynamic response. Those changes can be measured with methods like near-infrared spectroscopy (NIRS) [VPH⁺93] or functional magnetic resonance imaging (fMRI) [HSM09]. Although BCI control is possible using NIRS [SZG⁺07] as well as fMRI [WMB⁺04], both methods suffer from a bad temporal resolution.

For BCI control, a higher temporal resolution is preferred to allow a faster and more responsive control. This can be achieved by measuring the electrical activity of the brain, which in turn can be done using invasive as well as non-invasive recording methods. For the former the electrodes are placed, for example, directly on the surface of the brain, called Electrocorticography (ECoG). For the latter electrodes are placed on the surface of the scalp, called Electroencephalography (EEG). While invasive methods achieve a better signal-to-noise ratio as the signals are not damped by

the skull, they cause a high risk for tissue damaging or inflammations. For this reason, non-invasive methods should be preferred, at least if a sufficient signal quality can be achieved, which depends on the application purpose.

Recent works have shown that EEG is suitable for BCI applications, like spellers, wheelchair control, and others, which is why it is also used for the scope of this work and will be explained in detail.

2.2.1 Electroencephalography

The term Electroencephalography (EEG) was coined by Hans Berger, in 1929. Although Richard Caton was the first who found that the brain of apes and rabbits produce electrical currents [Cat75], Berger was the first who derived a human EEG [Ber29].

An EEG allows measuring the electrical activity on the scalp using electrodes which are often fixated on an EEG cap, whereby the electrodes should be positioned depending on the desired signals. Today, it is known that the electrical activity arises from the neurons in the brain which communicate mainly over electricity. Since there are approximately 86 billion neurons in the human brain [ACG⁺09], the EEG records a summation of electrical activity from a large group of neurons with a similar spatial location. As the skull damps the signals, the measured electric potentials have amplitudes in a range of only 10 μV to 100 μV [AGA⁺04], therefore, the sig-

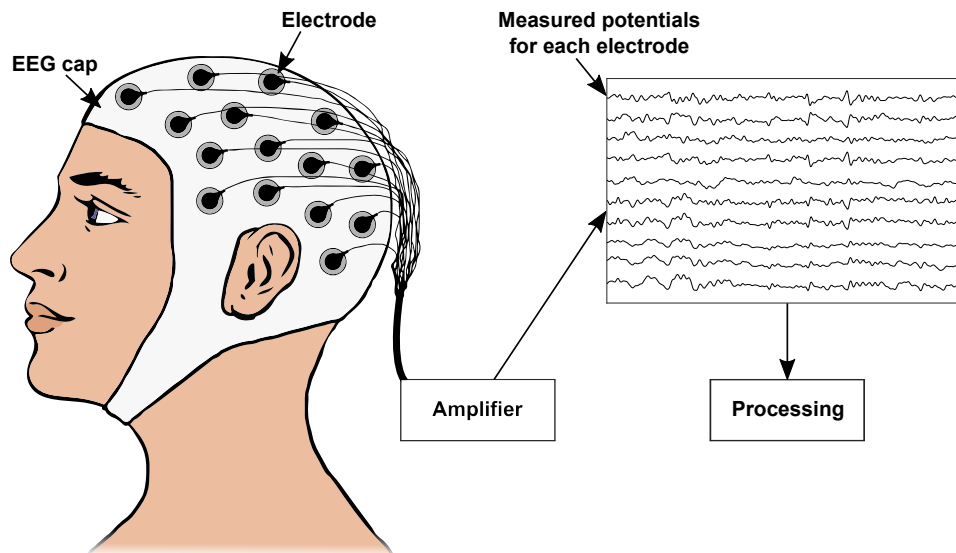


Figure 2.1: Sketch of how to record an Electroencephalogram. An EEG allows measuring the electrical activity on the scalp using electrodes which are often fixated on an EEG cap. For each electrode, the signals are amplified and can be used in the following for a desired processing.

nals must be amplified. For comparison, amplitudes using subdural electrodes are up to 10^3 stronger. A sketch showing an exemplary EEG setup is depicted in Fig. 2.1.

The human skull has a different shape and size for different subjects, therefore, standardized electrode positions were determined to get comparable results. Mainly the 10-20 [Jas58] or 10-10 [CLN85] system is used.

There are several types of EEG electrodes, which can mainly be distinguished into active/passive, and dry/gel/water based. Furthermore, the transmission can be wired or wireless, respectively. Today, active gel-based electrodes show to have the best signal-to-noise ratio, as an additional amplifier is placed within the electrode which makes the signal less prone to external influences, like noise caused by loose wires. Additionally, the usage of electrolyte gel increases the conductivity between the electrodes and the scalp.

Contrary, using gel electrodes requires time-consuming preparation and the gel dries out after a certain time. Dry electrodes can be used more easily and quickly, and it was recently shown that they are suitable for high-speed BCI control [XWP⁺18], albeit with a reduced performance compared to using gel electrodes.

2.3 Event-related potentials

To control a BCI, it must be driven by specific information that can be derived from the brain. EEG based BCIs are often based on event-related potentials (ERPs), which are defined as responses that occur after a specific, for example sensory, event and are further subclassified in *induced* potentials and *evoked* potentials [DKF06]. The latter are potentials which are time-locked to the stimulus and are further subclassified depending on the source of the event. For example, auditory [KFH⁺09, SK18] and somatosensory stimuli [MPSNP06, LRL⁺14] can be used to evoke potentials. Potentials based on visual stimuli have shown to achieve the best BCI performance and are used in this work. Therefore, they will be explained in detail in the following.

2.3.1 Visual evoked potentials

As mentioned, in 1984, Sutter stated that "the electrical scalp response to a modulated target is largest if the target is located within the central 1° of the visual field" [Sut84]. Depending on the type of stimulus, different responses (waveforms) can be measured. For clinical use, the International Society for Clinical Electrophysiology of Vision (ISCEV) defined standard protocols of how to evoke visual potentials, for example, to prognosticate eyes with a poor vision before planning surgery [OBB⁺16]. They defined three protocols with different types of stimuli, whereby two of them are often used for BCI control:

- **Pattern reversal VEPs** are evoked by using a checkerboard with white and black blocks. With a rate of 2 Hz the checkerboard is reversed, which means white blocks become black and vice versa. The resulting waveform is the most commonly used when VEPs are described literature. It has 2 negative and one positive wave (N75, P100, and N135 peaks), which appear at around 75ms, 100ms and 135ms after the stimulus, respectively. A typical waveform is depicted in Fig. 2.2.

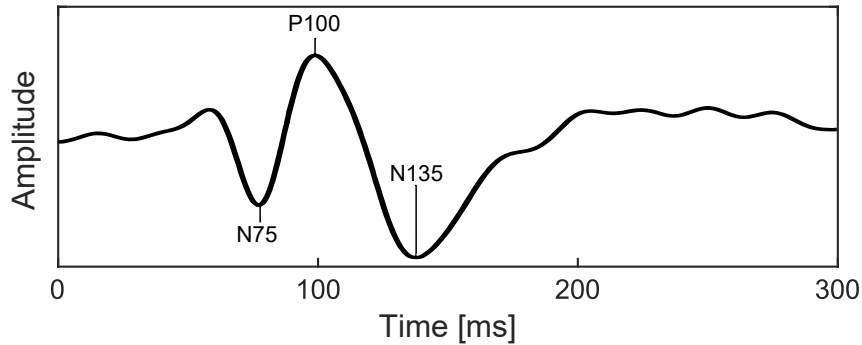


Figure 2.2: A typical pattern reversal VEP response. Vectorized version of Fig. 2 in [OBB⁺16]. Reprinted by permission from Springer Nature, © Springer-Verlag Berlin Heidelberg 2016.

- **Flash VEPs** are evoked using a white colored flash with a rate of 1 Hz, instead of using a checkerboard. The resulting waveform consists of multiple positive and negative waves, whereby P2, N3, and P3 and are the most prominent and occur at around 110 ms, 150 ms, and 220 ms, respectively. A typical waveform is depicted in Fig. 2.3.

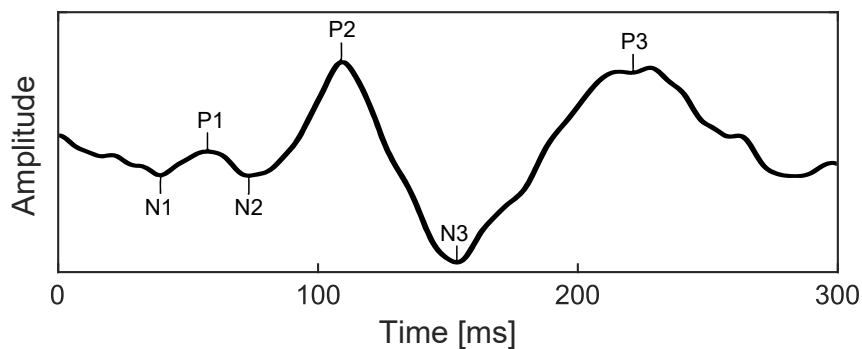


Figure 2.3: A typical flash VEP response. Vectorized version of Fig. 4 in [OBB⁺16]. Reprinted by permission from Springer Nature, © Springer-Verlag Berlin Heidelberg 2016.

Also, the shape and size of a stimulus influence the VEP response. Zerafa *et al.* [ZCFC13] and Waytowich *et al.* [WYK17] analyzed the differences for both, pattern reversal VEPs and flash VEPs.

Obviously, presentation rates of 1 Hz or 2 Hz, respectively, are not sufficient for multi-target BCI control. Therefore, several paradigms were proposed in recent years, which address this issue. The most common paradigms for BCI control will be explained in the following subsection.

2.4 Paradigms for BCI control

For BCIs based on ERPs, a possible command will be linked to a specific event/stimulus. Once the user focuses on a stimulus, the task is to classify the corresponding command. Until today, several stimulation paradigms were proposed for BCI control, whereby the most frequently used paradigms will be explained in the following. They are also used in the scope of this work.

2.4.1 Steady-state VEP

Steady-state visual evoked potentials (SSVEPs) are evoked by visual stimuli which are modulated with a certain frequency. The SSVEP is a periodic response of the brain, whereby the frequency spectrum of SSVEP contains the stimulation frequency and its harmonics.

In 1996, SSVEPs were first used for a BCI by Calhoun and McMillan, whereby the BCI had only one 13.25 Hz stimulus and users were tasked to self-regularize the SSVEP amplitude [CM96].

To use SSVEPs for multi-target BCI control, several frequencies and/or phase shifts must be used. When the user gazes one of the targets, a power increase can be found

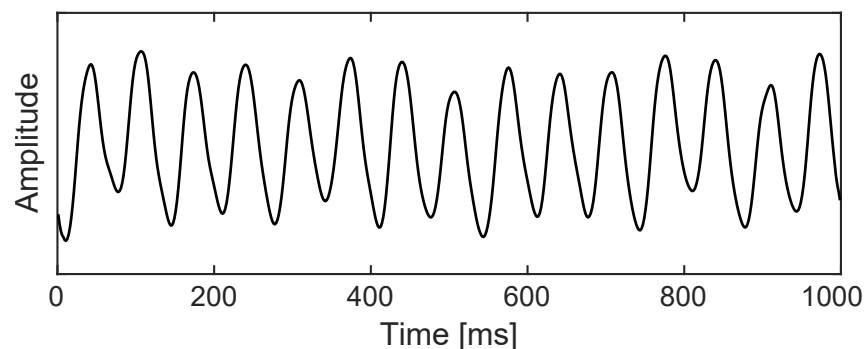


Figure 2.4: Typical SSVEP response using a visual stimulus modulated with a 15 Hz frequency. Shown are 1000 ms averaged over multiple trials.

in the same frequency range as the stimulation frequency. For example, using a 15 Hz stimulus results in 15 waves per seconds, this can also be seen in Fig. 2.4.

In 2002, Cheng *et al.* proposed the first usable, and for those times, fast SSVEP BCI with twelve possible commands [CGGX02]. In the following years, several frequency paradigms and classification methods were proposed, which increased the classification speed by a multi-fold [CWN⁺15].

It must be noted that the amplitudes of SSVEPs decrease for high frequencies and Herrmann [Her01] has shown that they can only be found in the EEG by using stimulation frequencies of up to 90 Hz.

2.4.2 Code modulated VEP

Sutter was the first who developed a BCI based on complex stimulation patterns [Sut84]. In 1992, Sutter extended his work and developed a BCI based on code modulated VEPs (cVEPs) by using pseudorandom codes with low auto-correlation, also known as m-sequences, for stimulation. The BCI had 64 targets and an ALS patient was able to spell 10 to 12 words per minute with a trial duration of 1.2 s. But it is worth to note that invasive electrodes were used. [Sut92]

Although Sutter achieved outstanding results for that time, his proposed stimulus paradigm has not been pursued for a few years. Instead, the previously explained SSVEP paradigm become popular. The firsts who picked up Sutter's paradigm were Hanagata and Momose in 2002 and 2007 [HM02, Mom07], but could not compete with the SSVEP paradigm. But Bin *et al.* has shown in 2009 and 2011, that a BCI based on cVEPs can achieve high-performance and outperformed all VEP BCIs at that time [BGW⁺09, BGW⁺11].

As mentioned, cVEPs are evoked by complex code modulated stimuli, an exemplary m-sequence will later be shown in section 4.3.1. Self-explanatory, such a complex stimulation pattern should also result in a complex VEP response. A typical cVEP response is depicted in Fig. 2.5.

2.4.3 P300

A P300, formally P3b, is a wave with a positive, strong amplitude with a peak that occurs approximately 300 ms after a rare task-related stimulus, for example, a user is guided to gaze a desired target until it flashes. It must be noted that the peak time varies between 250 and 500 ms [Pol07], which makes the detection more challenging. A typical P300 waveform is shown in Fig. 2.6, whereby it is averaged over multiple trials.

P300s can be used for BCI control, whereby they are often used for BCI spellers.

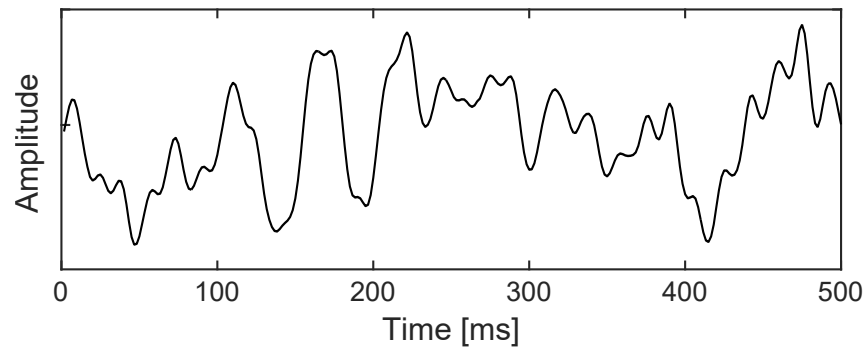


Figure 2.5: Typical cVEP waveform using a 63-bit pseudorandom code presented with 60 bit/sec. Shown are the first 500 ms averaged over multiple trials.

In 1988, Farwell and Donchin were the first who developed a P300 BCI speller. The letters (and additional commands) were arranged in a 6×6 matrix, whereby the rows and columns were flashed alternately in a random fashion for 100 ms with 500 ms inter-stimulus time. The user was tasked to count the flashes of the desired target. This procedure evokes P300s and allows to classify the corresponding row and column, which in turn are uniquely for a particular target. [FD88]

While the explained example is based on visual stimuli, P300 can also be evoked by other sensory stimuli, like auditory stimuli [KFH⁺09].

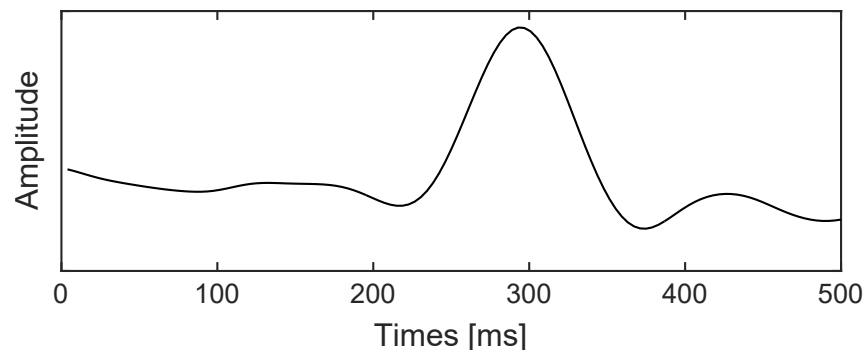


Figure 2.6: Typical P300 waveform. Shown are 500 ms averaged over multiple trials.

2.5 Machine learning

For BCI control, the measured brain signals are used. These signals must be analyzed suitably so that a particular target can be classified. For this, feature extraction is required which depends on the used stimulation paradigm. For example, for SSVEPs the power spectrum can be used, as it is known that a power increase can be detected in a specific frequency range depending on the stimulation frequency. Contrary, for other paradigms, like P300 or cVEP, the time domain is mostly used as a

feature. Regardless of the feature extraction, preprocessing steps are usually recommended to increase the signal-to-noise ratio, for example, spatial filters [SWRB14].

After feature extraction, those features must be assigned to the corresponding targets. Depending on the used stimulation paradigm, this requires a prior training phase or not. For example, for an SSVEP BCI, the powers of the corresponding frequencies can be used to classify the target, which does not necessarily require prior training [DMPL11, Cec10]. Contrary, cVEPs are evoked by complex stimulation patterns and can only be classified with prior training to differentiate them.

Machine learning is a computational way to solve such a task. In the past decades, several algorithms were developed to build mathematical models based on a set of training data. After training, those models can be used to make predictions of unknown data not used for training. In general, machine learning algorithms can be divided into supervised and unsupervised methods. The former requires a known output (label) for each input data point, and the task is to find a solution that maps the input data to the output data based on the training set. Contrary, unsupervised methods do not require a known output, and the task is, for example, to cluster the input data. For the current work, only supervised methods were considered, because labels are known.

It must be noted, that a "label" could be a discrete or a continuous value. The former is called a classification problem, the latter a regression analysis. Additionally, machine learning approaches are further differentiated into linear and non-linear approaches. The former enables only to recognize linear relationships, whereas the latter can also recognize non-linear relationships. Both linear and non-linear approaches are used in the scope of this work.

An important fact is to avoid over-fitting of the trained model. Over-fitting means, to "have unknowingly extracted some of the residual variation as if it represented model structure" [BA03]. The term "residual variation" means, for example, noise, which in turn is variance in the data that cannot be explained. Noise can be caused, in the context of EEG data, by external factors (like rays) or the general brain activity that is not task-related. This means, the prediction of an over-fitted model works well on the training data but may fail on additional data. To address this problem, several regularization approaches were developed.

In the following subsections, the machine learning approaches will be explained, which are used in this work.

2.5.1 Ridge regression

Regression in general tries to estimate the relationships among (continuous) variables. This is done by minimizing or maximizing a specific function, called cost function. Until today, several cost functions were proposed.

In this work, the ridge regression is used, which is a linear approach and was introduced by Hoerl and Kennard [HK70]. It is based on the Tikhonov regularization and tries to minimize the following cost function

$$\min_{\beta} ((y - X\beta)^T(y - X\beta) + \lambda\beta^T\beta) \quad (2.1)$$

where β is a vector of coefficients (weights), X is the input, called predictor, y is the desired output, called response, and λ is a regularization parameter.

The minimization of this cost function can be solved by

$$\hat{\beta} = \left((X^T X + \lambda I)^{-1} X^T y \right) / \sigma(X) \quad (2.2)$$

$$\beta_0 = \bar{y} - \bar{X} \hat{\beta} \quad (2.3)$$

where $\hat{\beta}$ are the coefficients, β_0 is a bias term, and I is the identity matrix. $\hat{\beta}$ consists of k coefficients $\beta_{1\dots k}$, one for each input dimension of X .

To predict the response of a new data point x , the following equation is used

$$y = \beta_0 + \beta_1 x_1 + \dots + \beta_k x_k \quad (2.4)$$

where y is the prediction for a data point $x = (x_1, \dots, x_k)$ with k dimensions.

2.5.2 Convolutional neural network

Although there are several non-linear approaches [BGV92, SW03], only a deep-learning approach is used in this work, because it was shown that such approaches outperform "traditional" machine learning approaches, especially for image classification and speech recognition [SZ14], but also for BCI control [KML17].

In general, deep-learning methods are "representation-learning methods with multiple levels of representation, obtained by composing simple but non-linear modules that each transform the representation at one level (starting with the raw input) into a representation at a higher, slightly more abstract level" [LBH15]. Combining multiple of those transformations allows learning complex mapping functions.

There are many different deep-learning approaches, in this work the so-called Convolutional Neural Network (CNN) is used. A CNN adapts the idea of receptive fields, which are biologically inspired. In the visual system, a receptive field of a single cell can be defined as "the region of retina (or visual field) over which one can influence the firing of that cell", whereby neighboring cells can have overlapping receptive fields [HW62]. For a CNN, this means that the output of an artificial neuron is influenced by a set of spatially adjacent input neurons.

2.5.2.1 Operations

As for other neural networks, a CNN generally consists of an input layer, (multiple) hidden layers, and an output layer, whereby the information flows uni-directionally from the input to the output, which is called feedforward. Each hidden layer transforms its input into an output through a specific operation. While there are many types of such operations, only those used in this work will be explained. The operation description is based on the Keras framework [C⁺15], which was used to implement the CNN.

Convolution A convolution operation is the basis of a CNN. As mentioned, the output of a neuron is influenced by a set of spatially adjacent input neurons. It must be mentioned that neighbored output neurons have neighbored/overlapping input neurons. This shows the peculiarity of CNNs, as they take the spatial structure of the input into account. The connections between those neurons are weighted, whereby the weight-matrix, called kernel, is the same for each output neuron and will be trained. The output h_i of a single output neuron with n input neurons can be calculated by:

$$h_i = \sum_{j=1}^n x_j \cdot w_j + b \quad (2.5)$$

where x_j is the input coming from neuron j and w_j is the corresponding weight of the connection.

A convolutional operation has multiple input parameters. The most important are `kernel_size`, `stride`, and `padding`. The former defines the spatial shape of the receptive field but it must be considered that the shape must not exceed the shape of the input. The kernel will then be moved over the input data, whereby the step size in each dimension is a parameter, called `stride`. This implies that the output shape of the layer depends on the stride, doubling it means halving the corresponding output dimension. Another important parameter is `padding`, which can be set to *same* or *valid* and defines what happens when the kernel reaches the edges of the input data. The former assures that the output shape is the same as the input shape, at least for a stride of 1. For this, the edges of each dimension of the input data will be filled up with zeros (depending on the kernel size) in order that the kernel can step beyond the edges. Contrary, *valid* causes no padding, which means that the output shape will be reduced depending on the kernel shape. In general, multiple kernels are trained for each convolutional layer.

Actually, only Conv1D operations are performed in the scope of this work, but in part on 2D input data, which requires to use the Conv2D operation.

Dense The dense operation is just a regular fully connected neural network layer. Contrary to the convolutional layer, the weights are unique for each connection, and the spatial structure is not considered. This also implies that the input shape, as

well as the output shape, is a vector. The output of a neuron is also calculated using Eq. 2.5, whereby the number of neurons is a definable parameter.

Activation Generally, the output of a neuron follows an activation function φ . Considering the output $\mathbf{h} = (h_1, \dots, h_N)$ for N output neurons:

$$\mathbf{h}_{\text{act}} = \varphi(\mathbf{h}) \quad (2.6)$$

The simplest activation function would be the linear function $\varphi(\mathbf{h}) = \mathbf{h}$, but various function types can be used. In this work, the rectified linear unit (ReLU) and softmax functions are used:

$$\varphi_{\text{ReLU}}(\mathbf{h}) = \max(0, \mathbf{h}) \quad (2.7)$$

$$\varphi_{\text{softmax}}(\mathbf{h})_i = \frac{e^{h_i}}{\sum_{j=1}^N e^{h_j}} \quad \forall i \in 1 \dots N \quad (2.8)$$

The ReLU function just removes negative values by mapping them to zero. The softmax function instead "squeezes" an N -dimensional vector to the interval $[0, 1]$, in other words, the input will be normalized into a probability distribution, whereby the sum over the N dimensions equals 1.

Batch Normalization Generally, a deep network, like CNN, consists of multiple layers, and the parameter optimization of each layer assumes that the output of prior layers does not change, but this is the case for each training step. Obviously, this slows down the training.

In 2015, Ioffe and Szegedy addressed this and introduced the batch normalization, with the results of a faster and more stable deep network training by removing the so-called *internal covariate shift* [IS15]. The batch normalization is generally applied after weight multiplication, but before applying the activation function φ , since the learning weights are causing the shift.

Pooling Pooling operations are used to reduce the number of dimensions. In the case of EEG, we have a high-dimensional space (samples \times channels), while the output is generally low-dimensional, for example, the number of classes. Pooling layers are similar to convolutional layers, whereby each output neuron has a receptive field. Contrary, no weights are trained, but the input is transferred to an output using a static function. Such a function takes the output of all input neurons in the receptive field and returns, for example, the average or the maximum of all values. In this work, only the latter is used.

As a pooling layer is similar to a convolutional layer, it also has the same parameters: `pooling_size`, `stride`, and `padding`, whereby the former defines the shape of the receptive field.

Dropout The dropout operation is a generalization method, which helps prevent over-fitting. It has one rate parameter, which defines a fraction of randomly chosen input units (connections) which will be set to 0, in other words, a random fraction of connections will be "eliminated".

Reshape Reshape is a simple operation which transforms a given input with a specific shape to another shape while maintaining the number of elements.

2.5.2.2 Hyperparameters

All parameters of the previously mentioned operations are called hyperparameters, but only a part of all hyperparameters was mentioned. They should be chosen with care as they can drastically affect the performance. But, tuning hyperparameters for deep neural networks is difficult, as there are many parameters which influence each other. A common method to tune parameters is *grid search*, which means the parameters will be varied in a certain range and the set of parameters, which resulted in the best performance, will be chosen. But, deep-learning approaches are generally computational expensive, and testing many sets of parameters can take a lot of time. Besides the above-mentioned operation parameters, the following parameters are important.

Learning rate As for other machine learning approaches, a CNN updates the weights by minimizing a cost function. For this, the so-called gradient descent is calculated, whereby the gradient is the derivative of the cost function. The minimization is an iterative process whereby the learning rate defines the step size of the weight adjustment, and the gradient defines the direction.

It must be noted, the general goal is to find the global minima of the cost function, but generally, there are multiple local minima. Using a lower learning rate can cause a stuck in local minima, whereby a larger learning rate could result in an overshoot of the global minima.

Batch size Instead of using the complete set of train data and only updating the weights once, several small subsets, called batches, could be used. Each batch will be propagated through the network, and the weights will be updated for each batch.

In general, the greater the batch size, the better the performance. Contrary, a large batch size leads to increased computational costs.

Number of epochs The number of epochs defines how often the entire training (propagating the complete train data) process will be repeated. Using fewer epochs could result in worse performance as the network could be further improved, which means the output is an under-fitted model. Contrary, using (too) many epochs requires more time and could result in an over-fitted model. In general, the number of epochs should be increased until the performance increase flattens.

2.6 Performance evaluation

To evaluate the performance of an approach, several measures can be used. Depending on whether a classification or a regression approach is used, various methods are suitable. Furthermore, to compare different approaches, not all methods are appropriate, because several factors must be considered. For example, to compare different BCI spellers, the number of possible targets, the classification accuracy, as well as required classification duration has to be taken into account. In the following, the used performance measures will be explained.

2.6.1 Accuracy

To evaluate the performance of a classification problem, a most commonly used method is to calculate the accuracy, which depends on the number of correctly performed classifications and the total number of performed classifications:

$$\text{ACC} = \frac{\text{number of correctly performed classifications}}{\text{number of all performed classifications}}. \quad (2.9)$$

It is important to note, that the accuracy does not take the number of possible commands (classes) nor the duration required for classification into account. Therefore, comparing the accuracy of one system to another makes only sense if both have an equal number of possible commands and an equal classification duration. Furthermore, the accuracy is only meaningful if each class was equally tested or if the predictor was trained using an equal number of data points for each class, respectively. Besides, the chance level must be considered.

2.6.2 Information transfer rate

The accuracy is not sufficient to compare the performance of different BCIs, as varying factors (like the different classification durations) are not taken into account. Today, the information transfer rate (ITR) is a commonly used method to compare BCIs. It was initially introduced by Claude Shannon in 1948 [Sha48], whereby it was first used by Wolpaw *et al.* in the context of BCIs [WRMP98].

The ITR estimates the information (bits) that can be transferred per time-unit and it can be calculated with the following equation:

$$\text{ITR} = \left(\log_2 N + P \log_2 P + (1 - P) \log_2 \frac{1 - P}{N - 1} \right) \cdot \frac{60}{T} \text{ [bit/min]} \quad (2.10)$$

where P is the prediction accuracy, T is the required time in seconds per classification and N is the size of the alphabet (number of classes).

As mentioned, the ITR is just an estimation, furthermore, it does not take corrections into account. But corrections are usually required if misclassifications occur, at least for a BCI speller. This was addressed by Dal Seno *et al.*, they proposed *Utility*, which can be seen as the ITR taking corrections into account [DSMM10].

It can be calculated using the following equation:

$$U = ((2P - 1) \cdot \log_2(N - 1)) \cdot \frac{60}{T} \text{ [bit/min]} \quad (2.11)$$

where the parameters are the same as for the ITR. Considering a BCI speller, the ITR will be 0 bit/min if the achieved accuracy equals the chance level. Contrary, the *Utility* will be 0 bit/min until an accuracy of 50% is achieved, as each misclassification requires an additional (correct) classification for correction.

2.6.3 Pearson Correlation

While the accuracy and the ITR can be used for classification approaches, they are not suitable for regression approaches. For this, a method is required which compares two (continuous) variables.

In 1895, Karl Pearson proposed the *correlation coefficient*, based on the work of Sir Francis Galton [Pea95]. It is a measure to express the degree of linear interrelationship between two variables. The value range lies within the interval $[-1, 1]$, whereby a value of 0 means that X and Y have no linear correlation. A value < 0 means the variables are negatively correlated, whereas a value > 0 means that the variables are positively correlated.

The Pearson correlation coefficient r between two variables X and Y can be calculated by:

$$r = \text{corr}(X, Y) = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y} \quad (2.12)$$

where $\text{cov}(X, Y)$ is the covariance between X and Y , and σ_X/σ_Y are the standard deviations of X and Y , respectively.

Squaring r results in the coefficient of determination r^2 , which is the proportion of variance in the dependent variable that is predictable from the independent variable.

Chapter 3

State of the art

3.1 Synchronous Brain-Computer-Interfaces

In 1977, Vidal described the first real-time synchronous BCI application [Vid77], which was based on single VEP responses. The stimuli were short xenon flashes for 30 μ s followed by 400 ms inter-stimuli time. In this study, he implemented a virtual maze which the user must go through, with four actions: up, down, left, and right.

Until today, several synchronous BCIs were developed, whereby most of them are based on SSVEPs, cVEPs, and P300s. The differences between the proposed methods are mainly the stimuli design, the feature extraction as well as the classification method.

3.1.1 P300

In 1988, Farwell and Donchin were the first who developed a P300 BCI speller [FD88]. The letters (and additional commands) were arranged in a 6×6 matrix, whereby the rows and columns were flashed alternately in a random fashion for 100 ms with 500 ms inter-stimulus time. The user was tasked to count the flashes of the desired target. This procedure evokes P300s and allows to classify the corresponding row and column, which in turn are uniquely for a particular target. For the classification, they used a step-wise discriminant analysis (SWDA) followed by a peak selection and a covariance evaluation. They stated that their P300 speller achieved an ITR of 12 bit/min with an accuracy of 95% and 26 s per classification, but by using Eq. 2.10, this results in only 10.6 bit/min.

In the following years, several approaches were made to improve the accuracy and/or the ITR [TGW06,GDS⁺09,JAK⁺04,HAAN13,KK14,CG11,SLS18,MHT17,SBK⁺12,PPS11,ZGW08]. For example, in 2006, Thulasidas *et al.* used a support vector machine (SVM) and achieved an average accuracy of 95% with 22 s

per classification, which corresponds to an ITR of 12.6 bit/min [TGW06]. Guger *et al.* used a linear discriminant analysis (LDA) but achieved only an ITR of 8.9 bit/min [GDS⁺09]. Those methods are generally based on averaging multiple P300s to increase the signal-to-noise ratio, as a single trial P300 detection have lower accuracies between 65% and 70% [JAK⁺04, HAAN13].

In 2014, Kaufmann and Kübler developed a P300 speller (6×6 targets), where they highlighted multiple two rows or columns simultaneously, instead of only one [KK14]. Furthermore, instead of increasing the luminance of a row/column, they used the face of Albert Einstein and the ying-yang sign. The stimuli were presented for 160 ms followed by 186.6 ms inter-stimulus time. For classification, they used a regularized linear discriminant analysis (rLDA). In an offline analysis, they showed to achieve an average ITR of 106.2 bit/min, which outperforms previous P300 BCIs by far.

Although recent deep-learning approaches have shown to further improve the performance [CG11, SLS18], P300 BCIs still lack in a relatively slow classification speed. Mainly because the P300 peak times vary between 250 and 500 ms [Pol07], and because the P300 amplitude decreases if the user gets used to the infrequent stimulus [RP99].

3.1.2 cVEP

As already mentioned, Sutter was the first who developed a BCI based on complex stimulation patterns [Sut84]. In 1992, Sutter extended his work and developed a BCI based on code modulated VEPs (cVEPs) by using pseudorandom codes with low auto-correlation, also known as m-sequences, for stimulation. The BCI had 64 targets, and an ALS patient was able to spell 10 to 12 words per minute with a trial duration of 1.2 s. But it is worth to note that invasive electrodes were used. [Sut92]

The basic idea of a cVEP BCI is to modulate several targets with the same stimuli pattern but shifted. Afterward, the goal is to detect the corresponding shift, which is unique for each target. This is possible as stimulation patterns with low auto-correlation are used.

Although Sutter achieved outstanding results for that time, his proposed stimulus paradigm has not been pursued for a few years. In 2002 and 2007, Hanagata and Momose were the first who picked up Sutter's paradigm and showed that it can be used with EEG electrodes, but they achieved low ITRs of <20 bit/min [HM02, Mom07].

In 2009 and 2011 Bin *et al.* were the first who achieved a high-spelling performance [BGW⁺09, BGW⁺11]. They outperformed all recent VEP BCIs and achieved ITRs of 108.1 bit/min. The performance improvement was mainly achieved by using a spatial filter based on a canonical correlation analysis (CCA). In an initial train phase, the users had to gaze a reference target for multiple trials, the corresponding

spatially filtered EEG was averaged and acted as a template. For each target, the template was shifted according to the shift of the stimulation pattern. During the test phase, the target classification was done by calculating the correlation coefficient between the spatially filtered EEG and all templates, the corresponding target whose template correlates most was selected.

The method was further improved by Spüler *et al.* using a one-class support vector machine (OCSVM) instead of averaging the spatially filtered EEG [SRB12a]. In combination with an adaptive classification algorithm based on error-related potential (ErrPs) they achieved an average online ITR of 144 bit/min. [SRB12b]

3.1.3 SSVEP

In 1996, SSVEPs were first used for a BCI by Calhoun and McMillan, whereby the BCI had only one 13.25 Hz stimulus, and the users were tasked to self-regularize the SSVEP amplitude [CM96]. Until today, many BCIs based on SSVEPs were developed [CGGX02, Vol11, CWW⁺11, MCVH12, DMPL11, PMTA09, ZTLZ17, XWP⁺18, YCW⁺15, ZXCY14, HLJ⁺12, Cec10, TZ15, ZXL⁺12, CWN⁺15].

In 2002, Cheng *et al.* proposed the first usable, and for those times, fast SSVEP BCI with twelve possible commands and achieved an average ITR of 27.15 bit/min [CGGX02]. They used a fast fourier transformation (FFT) to calculate the power frequencies in the measured EEG. For each stimulation frequency (between 6 Hz and 14 Hz), the power of the frequency as well as for the second harmonic were summed up. If the power was over a certain threshold (twice the mean of the amplitude spectrum between 4 Hz and 35 Hz), the corresponding target was selected. This implies that their method does not require a prior training phase.

As mentioned in section 2.4.1. The SSVEP is a periodic response of the brain, whereby the frequency spectrum of SSVEP contains the stimulation frequency and its harmonics. Especially the latter makes frequency coding for multiple targets quite challenging, as harmonic frequencies should be avoided. Furthermore, as most standard computer monitors are used for stimuli presentation, the number of usable frequencies is further limited. For example, using a frame rate of 60 Hz allows a maximum stimulation frequency of 30 Hz, additionally, the stimulus frequency should be a whole divisor of the frame rate.

In 2011, this was addressed by Ivan Volosyak, who developed an SSVEP BCI speller with only 5 targets, whereby four of them allow navigating through the 32 letters/symbols of the keyboard. To spell the selected letter, the fifth target is used. The advantage of this approach is that fewer targets are used, which makes the classification easier. Contrary, spelling a letter requires to perform multiple classifications, which reduces the spelling speed. Nevertheless, this approach achieved an average ITR of 61.7 bit/min. [Vol11]

In the same year, Cao *et al.* addressed the same issue with another approach. Instead of using navigation targets, they used 3 different 15 target layouts, whereby two of these targets were used to switch between layouts. In total, this allows for spelling 42 different letter/symbols. With an average ITR of 61.6 bit/min, they achieved an equal spelling performance. [CWW⁺11]

All methods mentioned so far, make use of frequency coding, which means each target is modulated with its own stimulation frequency. Contrary, also phase coding can be used, which means several targets are modulated with the same stimulation frequency, but with varying phase shifts. For example, Manyakov *et al.* developed an SSVEP BCI with 6 targets, all modulated with a 10 Hz stimulus, but successive targets are phase-shifted by $\pi/3$. They achieved an average ITR of approximately 25 bit/min. [MCVH12]

In general, the stimulus coding and target identification methods play important roles in the performance of BCIs based on SSVEPs. In 2015, Chen *et al.* [CWN⁺15] have shown that a complex stimuli design combining multiple frequencies and phase shifts, called joint frequency-phase modulation (JFPM), enhances the discriminability between SSVEPs with a very narrow frequency range. As mentioned above, this is the most challenging conditions in frequency coding. They implemented a 40-target BCI speller which achieved an average online ITR of 267 bit/min and drastically reduced the required time for a classification, compared to all recent non-invasive BCIs. But, as mentioned in the previous chapter, their method is only suitable for experienced users, as the time between classification and the next stimulus amounts only 280 ms.

3.2 Asynchronous Brain-Computer-Interfaces and non-control state detection

As explained in the previous chapter, synchronous BCI spellers are not suitable for real-world applications, because if the user needs too much time to think about the following word, the BCI will not wait and will classify a random command. Therefore, a practical BCI should be asynchronous, or also called self-paced. As mentioned, the term "asynchronous" is not uniformly defined for BCIs, sometimes it is used for early stopping methods disregarding the non-control (NC) state detection and sometimes for distinguishing between IC and NC. It is also worth to note that no state-of-the-art criteria exist what a user should/could do during the NC state, which makes the comparability difficult.

Some methods make use of hybrid BCIs combining several brain activities, for example, using ERPs (like P300) to distinguish between IC state and NC state in combination with steady-state VEPs (SSVEPs) for classification [PPS11,LPWY13]. Others use threshold methods, like Cecotti [Cec10] who developed an asynchronous

SSVEP BCI distinguishing between IC state and NC state by normalizing frequency powers for each stimulus frequency following a minimum energy combination approach, and the frequency is detected if the normalized frequency power is greater than a predefined threshold, otherwise an NC state is detected. Another approach was proposed by Suefusa and Tanaka [ST18] who developed an asynchronous SSVEP BCI using multiset canonical correlation analysis (MCCA) and a multi-class support vector machine to distinguish between 28 IC classes and the NC class. However, compared to synchronous methods which achieve ITRs >100 bpm, the current asynchronous BCIs are substantially slower, with 67.7 bpm being the fastest asynchronous system [ST18] to date.

Furthermore, different approaches addressing the NC state detection are also not directly comparable. On the one hand, there are trial-based approaches with fixed trial durations classifying each trial as IC or NC, on the other hand, there are continuous classifications without fixed trial durations. For the former, it is possible to specify the NC detection accuracy and recent works [ASA⁺11, PPS11, PFH⁺15, MNK⁺17, PMTA09, XLX⁺13] achieved accuracies between 76.94% and 98.91%. For the latter, it is possible to specify the number of erroneous classifications during the NC state per time unit, and recent works [ZGW08, LPWY13, ABG18] achieved 0.7 to 0.49 erroneous classifications per minute. In summary, all of them did not achieve a reliable NC state recognition, meaning the BCI executes random commands during the NC state, which decreases the user experience.

3.3 Home-use Brain-Computer-Interfaces

As mentioned, one of the main purposes of BCIs is to restore communication of motor disabled people, for example, patients who suffered a stroke or have amyotrophic lateral sclerosis (ALS). Obviously, a user would intend to use the BCI at home and not only in a lab. But recent BCIs are not really suitable for home-use, either they are based on synchronous control or asynchronous control with low spelling performance and insufficient NC state detection.

All previously mentioned BCIs were evaluated in a lab and mainly with healthy subjects. Until today, only a few works addressed the home-use of BCIs with disabled users [VMS⁺06, SVW10, MMV⁺12, MSM⁺14, MVD⁺15, HBKK15].

In order that a BCI system can be used at home, it must fulfill some criteria. Miralles *et al.* and Holz *et al.* have defined some such criteria [MVD⁺15, HBKK15]. Miralles *et al.* also asked users to rate criteria by priority, according to that, the most important criteria are effectiveness, reliability, and robustness. Obviously, a BCI has no real benefit if tasks cannot be performed with a certain accuracy and if the BCI system does not work reliably. Further important criteria are functionality (as many different tasks as possible), ease of use, efficiency (communication speed), comfort,

and mobile usage, in that order. One of the most challenging criteria is robustness, optimally, a home-use BCI should work for several years without malfunctioning.

All recently evaluated home-use BCIs are based on P300 ERPs. For example, Sellers *et al.* developed a system with two different layouts, one allows for spelling, and the other one allows environmental control, like controlling a TV. They evaluated their system for a period of 2.5 years with a 51-year-old man with advanced ALS. A user's caregiver had to learn to operate the system, which includes placing/preparing the EEG gel electrodes and starting the software. The system was remotely calibrated 1-2 times per week, and the caregivers made notes regarding system usage and performance changes. The results revealed that the P300 responses have remained stable in magnitude and form for the whole period. Also, the classification performance remained stable at around 83% (18.5 bit/min). They also noted that the user was satisfied with the system as it enables him to communicate with his family and improved his independence. [SVW10]

Another long-term study is by Holz *et al.* [HBKK15], they developed a home-use BCI based on the *Brain Painting* framework [ZHK⁺13]. The system was used by a 73-year-old female ALS patient for 14 months. Within that period, approximately 200 paint sessions were performed by the user. After each session, the user was asked to give a subjective rating with justification for the following criteria: effectiveness, efficiency, satisfaction, and quality of life. The results reveal that the system had "a positive impact on happiness, self-esteem, [...], self-confidence, and ability to participate". Furthermore, the user stated that the system makes her "happy and free". But the authors also point out, that the system cannot be used independently, the user has to wait for a third person to set up the BCI. [HBKK15]

The BCI systems proposed by Mak *et al.* [MMV⁺12] and McCane *et al.* [MSM⁺14] were also evaluated with ALS patients. Although the systems were used at the user's home, the systems were only used for a single day and were assisted by an expert.

Altogether, these results show the importance of home-use BCI, as they significantly improve the quality of life for affected persons. But those systems must be further improved to increase the performance and to reduce the dissatisfaction, for example, caused by misclassifications.

3.4 Brain response prediction to visual stimuli

Until today, there are virtually no methods which are able to predict the brain response to visual stimuli. To the best of my knowledge, there are only three notable works which will be introduced in the following.

Recently, Sobreira *et al.* [STK18] developed a deep-learning model which simulates the visual pathway. Their model was designed in three layers, whereby those layers

should represent the functionality of several cells which are involved in the visual pathway. But it must be mentioned that the model is not entirely biologically plausible. The input of the model are images with a size of 640×640 pixels, and the outputs are 400 spiking neurons representing the predicted SSVEP, whereby the model has a temporal resolution of 0.01 ms. They evaluated their model using differently shaped stimuli, like checkerboard, squares, circles, and others. The different shapes were presented for 200 ms followed by a black image for 600 ms. Unfortunately, only results for 4 of those stimuli were given, which resulted in a median $r^2 = 0.828$. But it must be emphasized that the method was not evaluated for complex stimulation patterns which result in complex VEPs but only for single stimuli.

Another approach was developed by Wang *et al.* [WGG15]. They developed a computational model which can predict (only) SSVEPs for a given frequency. Besides the frequency, the model is based on several parameters, like the number of harmonics, the amplitude strength, and the phase shift, respectively. They proposed that their model can be used for the design and implementation of a multi-target SSVEP-based BCI. For example, by analyzing predicted SSVEPs for varying parameters, they found that the harmonic SSVEP components could provide valuable information for frequency detection. Unfortunately, they neither compared the model predictions to real measured SSVEPs, nor they evaluated if their findings really improve the classification performance. Therefore, the model prediction should be interpreted with caution. Furthermore, as shown in Fig. 2.4, an SSVEP response shows a (more or less) static course, which is obviously easier to predict as complex VEPs, like cVEPs (see Fig. 2.5).

One notable exception investigating a more general approach to use visual stimulation that is not restricted to predefined patterns is the work by Thielen *et al.* [TvdBFD15]. Their model assumes that the response to a complex stimulation pattern is a linear superposition of individual single-flash VEPs [CPAD⁺11]. Thielen *et al.* therefore developed a convolution model that breaks down a complex stimulation pattern in smaller sub-components and models the response to a complex stimulation pattern by a superposition of the responses to the sub-components. While this approach allows for a more flexible prediction model, the stimulation patterns are not fully arbitrary, as they can only be composed of short and long pulses. Furthermore, although their method should be able to predict the brain response to random compositions of short/long pulses, they only evaluated the performance using a small subset of 65 stimulation patterns with 120 bits each. Unfortunately, although they used the method for BCI control, they did not analyze the brain response prediction of their model on a single trial basis but averaged over multiple trials, which resulted in $r^2 = 0.476$.

Chapter 4

Visual stimulation framework for high synchronicity

In the field of visual neuroscience as well as for BCIs, experiments based on visual stimuli are often required to have a strict timing on a millisecond scale. For example, if an experiment presents visual stimuli and the subject has to push a button to measure the reaction time, it's required to know the exact timing of both the stimulus and the button press, generally, this is done by storing timestamps. If there are any latencies, they should be corrected, otherwise, the results will be distorted leading to wrong conclusions like measured reaction times are longer as they really are.

Especially for VEP based BCIs, it is required to know the exact timings of stimuli presentation, as the brain responds in a millisecond scale. If stimuli timings vary, VEPs will be time-shifted corresponding to that variation. Recent studies have shown that latencies of P300 event-related brain potentials (ERP) and error-related potentials (ErrPs) vary depending on the experiment [GP02, ICM⁺14] and that correcting latencies leads to a better generalization [ICM⁺14] and an increased performance [MHT17].

As shown by Wilson *et al.* [WMSW10] a PC has several potential factors which cause latencies, for example, the operating system latency, the video output latency, or the monitor input lag. Since most experiments use PCs for stimulus presentation and data analysis, those latencies should be considered. They also showed that several other factors exist especially for BCIs. They used BCI2000 [SMH⁺04] a general-purpose software system for BCI control and measured latencies caused by the amplifier, the software signal processing, and other factors. If the factors which cause more or fewer static latencies are known, they can be corrected easily by fixing the timestamps or by shifting the data, respectively. Contrary to static latencies, varying latencies (jitter) can dramatically alter the results and are harder to handle, as it is required to know how latencies vary and the data must be corrected accordingly. If the jitter is not corrected, it could lead to a distortion of results.

In this chapter, the Simplified Application for Visual Experiments (SAVE) framework is introduced. It allows to easily create presentation layouts with arbitrary stimulation paradigms. Furthermore, it can be controlled through TCP/IP and assures exact synchronization of the visual stimuli with an EEG amplifier.

4.1 General description

The SAVE framework is implemented in MATLAB [MAT17] and is based on the Psychtoolbox-3 [BV97], which in turn is based on the OpenGL[®] application programming interface. In general, it is a framework for experiments based on visual-evoked potentials (VEPs) and event-related potentials (ERPs). Among the predefined layouts and stimulation patterns, it offers an interface to easily build presentation layouts with freely placeable targets (see section 4.4) and an interface to create arbitrary stimulation patterns for each target of the layout (see section 4.3).

The stimuli presentation is bound to the refresh rate of the monitor and the presentation speed (number of frames per stimulus) can be defined. Since visual experiments require strict stimuli timings, the framework also takes frame drops into account and compensates them by informing the stimulation interface about the number of recent frame drops, if any occur, and correct them if desired. Furthermore, it also takes the monitor raster latency (see chapter 5) into account by offering information on where each target is vertically placed on the screen, to easily correct the latencies.

The application is split into two modules communicating over TCP/IP. One module is responsible for the experiment itself and one to remotely operating the experiment (set parameters, choose targets, etc.). At each frame build, the operator module will get information about how each target is modulated. The synchronization with an EEG amplifier is done by the parallel port by giving information about the current state (see section 4.2).

The framework was tested on Microsoft[®] Windows 10[™] but should work on Linux and Apple[®] macOS[™] with some slight modifications. A detailed technical documentation, including how to create layouts/stimulation patterns and how to design an experiment, can be found online [Nag19].

4.2 Synchronization with an EEG amplifier

The synchronization is realized using the parallel port as it has a near-to-zero latency and it can be used with several EEG amplifiers to synchronize it with the recorded EEG data. The 8 bits contain the following information:

- **1. bit:** is 1 if a run has started, 0 otherwise.

- **2. bit:** is the stimulation pattern of the first target, this can be used to check if the stimulation pattern is as expected.
- **3. bit:** toggles between 1 and 0 for each drawn frame, this can be used to detect frame drops and to synchronize the known stimulation patterns with the sampling rate of the amplifier.
- **4. bit:** is 1 during a trial and 0 otherwise.
- **5. - 8. bit:** can be set dynamically for each implemented stimulation pattern. For example, to mark the start of a sequence if a cyclic stimulation pattern is used.

Therefore, the exact time point of a trial start/end, as well as the precise timing of every single stimulus, can be determined.

4.3 Stimulation paradigms

Several widely used stimulation paradigms are already implemented in the framework, additional paradigms can easily be implemented as explained in the documentation. For the scope of the current work, the following stimulation paradigms are used.

4.3.1 cVEP

This is a stimulation paradigm which uses a stimulation pattern with low auto-correlation, also known as m-sequences. Those patterns are generated using the code of Buračas and Boynton [BB02], which has 4 parameters (`baseVal`, `powerVal`, `shift`, `whichSeq`) to vary the sequence. The generated m-sequence is assigned to each target, but bit-wise shifted for successive targets, whereby the shift can be defined and depends on the number of targets of the corresponding layout. The values, used in the following experiments are (2,6,shift,1). With `shift=1` this results in the following 63-bit m-sequence:

```
101011001101110110100100111000101111001010001100001000001111110
```

4.3.2 SSVEP

This stimulation paradigm is based on frequencies, which means the single stimuli are presented in a sinusoidal fashion, whereby the corresponding frequency and phase-shift can freely be defined for each target of the layout. Here it is worth to mention that the framework allows a binary stimulation (on/off) or stimulation with

sinusoidal transitions between on and off. Furthermore, it should be considered that the stimulation frequencies are limited by the refresh rate of the monitor. This stimulation paradigm is used in chapter 5 and 15 with binary stimulation.

4.3.3 Random

This is a completely new stimulation paradigm introduced in this work. The MT19937 [MN98] random generator is used to generate a fully random modulation pattern for each target. The random generation can be varied by the seed parameter of the random generator.

4.4 Layouts

Depending on the application purpose, a layout can be created with freely placeable targets. Each target has two states, one which is shown as long as the stimulus is 1 and another one which is shown as long as the stimulus is 0. The size, shape, and color can be set for every single target. As an alternative to a color, also a texture can be set for each target. Additionally, each target can have a label whose color and font size can freely be defined. In this work mainly the following layouts are used.

4.4.1 Matrix keyboard

A matrix keyboard layout is often used for BCI spellers, as the targets are equal sized and can be distributed evenly across the screen. In this work, a 4×8 matrix keyboard layout (32 targets in total) was built, whereby the targets have a rectangle shape are labeled alphabetically from A to Z followed by ' _ ' and numbers 1 to 5. The targets are colored in white (binary 1) and black (binary 0) and are separated by a blank black space. Additionally, there is a text-field above the targets showing the written text. A screenshot of the layout is shown in Fig. 4.1.

4.4.2 QWERTZ keyboard

Although a matrix-keyboard with lexicographic order has advantages for scientific purposes like equal sized targets, most participants/end-users are familiar with established keyboard layouts. Therefore, also a 55 target German QWERTZ-layout was built. The layout has a CAPS (⇩) and SHIFT (⇧) targets which allows writing case-sensitive letters as well as all standard symbols and punctuation marks. Like for the matrix layout, a text-field was placed above the targets. A screenshot of the layout is shown in Fig. 4.2.

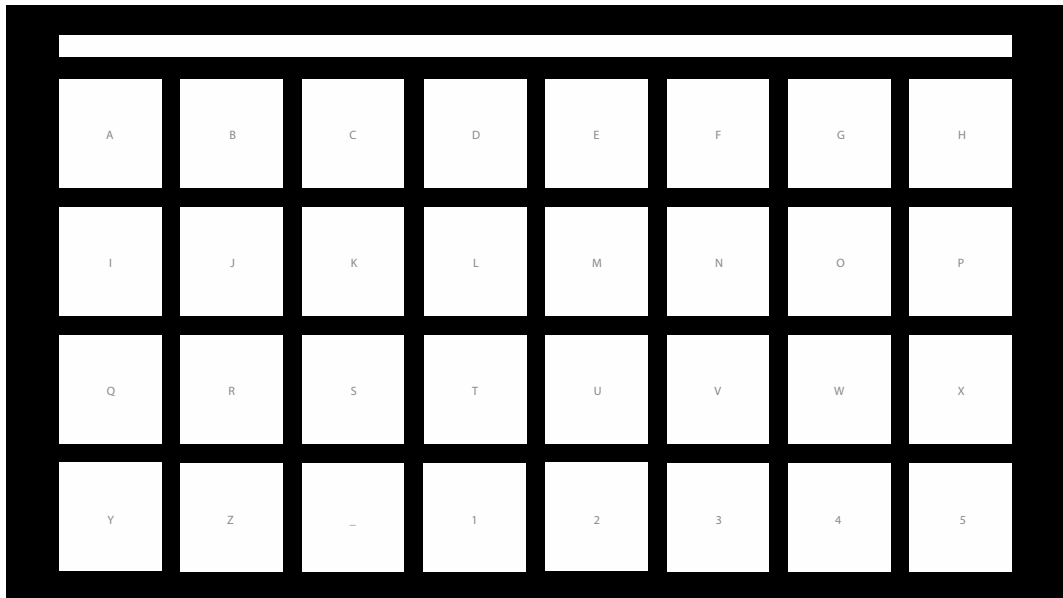


Figure 4.1: Matrix keyboard layout with 32 targets.

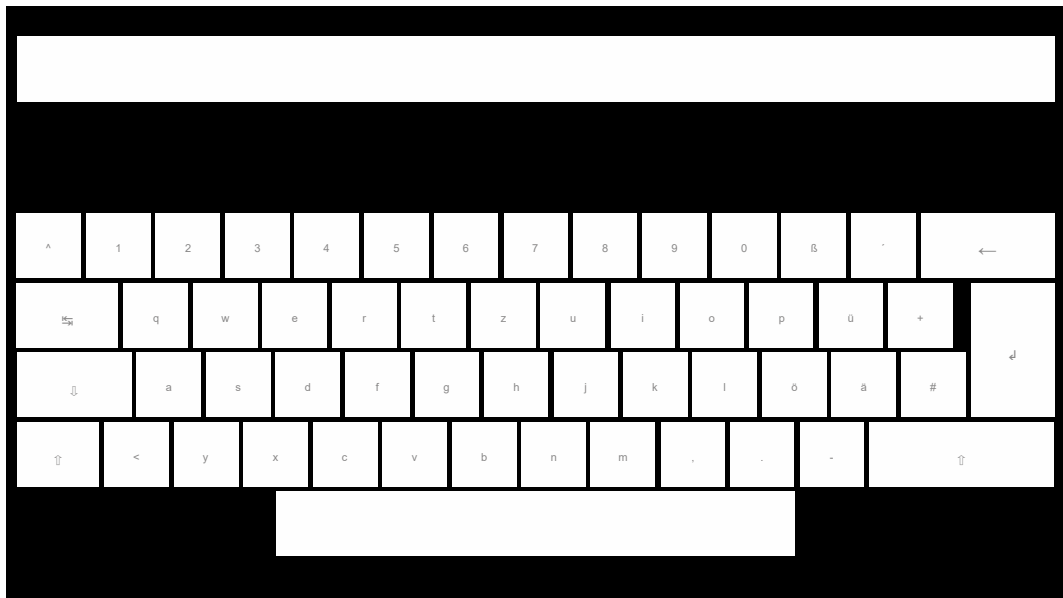


Figure 4.2: German QWERTZ keyboard layout with 55 targets.

Chapter 5

Monitor raster latency and its effect

As explained in chapter 4, for VEP based BCIs it is required to know the exact timings of stimuli presentation, as the brain responds in a millisecond scale. Especially varying latencies (jitter) should be corrected, otherwise, it could dramatically alter the results. One jitter-causing factor is mostly unconsidered: the monitor raster latency. Experiments using a monitor for stimulus presentation should consider the fact of how a monitor will present every single frame. A frame will be presented line-wise from top to bottom, resulting in an increased latency from the upper left pixel to the bottom right pixel, the raster latency. More precisely, the latencies are based on the addressing scheme of the monitor. For example, a cathode ray tube (CRT) monitor presents the pixels from left to right and top to bottom. Furthermore, a CRT scan can be un-interlaced (first line to last line) or interlaced (odd lines first followed by even lines). Liquid crystal displays (LCDs) and organic light emitting diode (OLED) displays generally present a frame line by line [PSD09]. Regardless of the display technique, the total processing time of each frame is approximately 95% of the inverse of the refresh rate. For a refresh rate of 60 Hz, there will be a delay of $0.95/60 = 15.8\bar{3}$ ms between the first and the last pixel of that frame, as shown by Tobias Elze [Elz10]. This property of a monitor, that leads to varying latencies, are attended by some researchers in the field of neuroscience [GVLdIM14] but doesn't seem to get much attention in visual BCI experiments, although all BCIs using a standard monitor will be affected by the raster latency regardless of whether they are based on cVEP, SSVEP, or ERP. Interestingly, in 1992 Sutter proposed the "scan delay" of a CRT monitor which resulted in slightly asynchronous stimuli depending on the vertical position of the stimulus [ST92]. This is exactly the described raster latency, but no further attention was given to that by others.

In this chapter, the effects of the monitor raster latencies on SSVEP, cVEP, and P300 are shown. Furthermore, a method is proposed to correct the raster latencies, and it is shown that the BCI performance can be significantly improved. Parts of the results, figures and wording are already published in [NDRS18].

5.1 Methods

5.1.1 Measuring raster latencies

To determine the raster latencies, an old CRT monitor (Iiyama A901HT), an old LCD monitor (Dell 1908FPc) and a new LCD monitor (BenQ XL2430-B) of the year 2016 using the latest technology with low reaction times, were measured. The presentation layer was implemented using the framework of chapter 4. A stimulus was presented once each second on the full screen-size for the length of one refresh cycle of the monitor. Since the refresh rate was set to 60 Hz, each stimulus had a length of $16.\bar{6}$ milliseconds. The parallel port was used as the trigger and was set right after the `Screen('Flip', ...)` command, which should - theoretically - present the stimulus immediately at the start of a refresh cycle if there were no latencies at all. The time at which the stimulus was presented on the monitor was determined by a photodiode which was held once at the top left and once at the bottom right position of the monitor. To measure the timings an oscilloscope (Rohde&Schwarz HMO1022) with a sampling rate of 25 kHz was used. All measures were repeated 5 times for each monitor.

As the measured voltage of the parallel port switches immediately between states, it is easy to determine the onset time which represents the theoretical stimulus onset time. The monitors need a specific amount of time till full illumination is reached, because of this and the fact that the photodiode has a small jitter the real stimulus onset time of the monitors was specified as the time point at which 100 successive samples (4 ms window) are above the mean baseline.

5.1.2 Analysis of SSVEP data

A simple SSVEP experiment was implemented to determine the effects of the measured raster latencies in the brain's response.

5.1.2.1 Setup

The setup consisted of an g.USBamp (g.tec, Austria) EEG amplifier, a PC and the LCD monitor (BenQ XL2430-B) mentioned above. The presentation of the stimuli was operated from the PC and synchronized with the EEG amplifier by using the parallel port. BCI2000 [SMH⁺04] was used as a general framework for recording the data of the 32 electrodes, from which 30 were located at Fz, T7, C3, Cz, C4, T8, CP3, CPz, CP4, P5, P3, P1, Pz, P2, P4, P6, PO9, PO7, PO3, POz, PO4, PO8, PO10, O1, POO1, POO2, O2, OI1h, OI2h, and Iz. The remaining two electrodes were used for electrooculography (EOG), one at the nasal bridge and one at the outer canthus of the left eye. The ground electrode (GND) was positioned at FCz and reference

electrode (REF) at OZ. The monitor refresh rate was set to 60 Hz and the amplifier sampling rate to 600 Hz, resulting in 10 samples per frame.

5.1.2.2 Experimental design

The stimuli were presented at the top left and bottom right area of the monitor, to evaluate the full magnitude of raster latencies caused by the monitor. The framework of chapter 4 was used to present a 5 cm×5 cm square to the subject with a stimulation rate of 1 Hz and 15 Hz, respectively. Each stimulus was presented for one frame, resulting in a stimulus length of 16.6 milliseconds.

To avoid fatigue, a run consists of 4 parts with 2 minutes each: (1) 1 Hz top left position, (2) 1 Hz bottom right position, (3) 15 Hz top left position, and (4) 15 Hz bottom right position. In total the subject had to perform 3 runs, therefore, this results in 6 minutes of EEG data for each stimulation frequency.

5.1.2.3 Processing

The EEG data was notch-filtered by the amplifier at 50 Hz and additionally to increase the signal-to-noise ratio a 200th-order bandpass finite impulse response filter was applied between 0.1 Hz and 30 Hz. To avoid a phase-shift due to the filtering, the MATLAB `filtfilt` function was used, which performs zero-phase digital filtering.

The EEG data of electrode O2 was analyzed by averaging over windows of 1-second length, resulting in $6 \cdot 60 = 360$ windows for each of the 4 parts. To determine the time-shift between the top-left and bottom-right position, the cross-correlation was used. This results in the number of shifted samples at which the VEP responses correlate most, which in turn can be converted to the time-shift.

5.1.3 Analysis of P300 data

5.1.3.1 Setup

The P300 data used for latency estimation is from [SBK⁺12], in which 24 subjects (18 healthy, 6 motor impaired) used a P300 BCI speller in 2 sessions. For the P300 speller, the BCI2000 implementation was used, and the speller was presented on a 19 inch TFT monitor. The EEG data was recorded at a sampling rate of 256 Hz with a g.USBamp (g.tec, Austria) EEG amplifier from 16 passive EEG electrodes placed at F3, Fz, F4, T7, C3, Cz, C4, T8, Cp3, Cp4, P3, P4, Po7, Po8, and Oz. More details about the setup can be found in [SBK⁺12].

5.1.3.2 Processing

The EEG data was bandpass filtered by a 4th order Butterworth filter between 0.5 Hz and 12 Hz. To visualize the P300, the average of all non-P300 trials was subtracted from the average of all P300 trials, separately for each session of each subject. This was done for the symbol in the top left of the speller matrix and the symbol in the bottom right. To compute the latency between the two, a cross-correlation analysis was performed.

5.1.4 Analysis of cVEP BCI data

Using the data from [SRB12b] an offline analysis was performed to compare the performance differences of correcting the raster latencies versus not correcting.

5.1.4.1 Setup

The system consisted of an g.USBamp (g.tec, Austria) EEG amplifier, a PC and the CRT monitor (Iiyama A901HT) mentioned above. Stimulus presentation and online classification were operated from the PC. The presentation of the stimuli was synchronized with the EEG amplifier by using the parallel port. The data was recorded using BCI2000 [SMH⁺04] with a sampling rate of 600 Hz. The visual stimuli (black or white) were presented on the monitor with a refresh rate of 60 Hz and a resolution of 640×480 pixels. DirectX (Microsoft Inc.) was used to ensure synchronization of the presented stimuli with the refresh rate of the monitor. The presentation layout had 32 targets (8×4 grid) surrounded by 28 non-targets and forms a 10×6 grid. For modulation of the stimuli, a 63-bit binary m-sequence was used, because of its low autocorrelation property. For each target, the same sequence was used for modulation, but circular-shifted by 2 bits for each successive target.

5.1.4.2 Experimental design

Each trial had a length of 1.05 seconds followed by 0.85 seconds for gaze shifting. In total each of the 9 subjects had to perform 64 trials for training and 576 trials for testing, split into two sessions.

5.1.4.3 Processing

The measured EEG data was processed as described in [SWRB14]. In short, the data was spatially filtered by a canonical correlation analysis (CCA) and rather than averaging the EEG data from multiple stimulation sequences, a one-class support vector machine (OCSVM) was used to estimate the probability distribution of the

data. The center of the OCSVM hyper-sphere was used as a template, which was shifted to obtain templates for all targets. By calculating the Euclidean distance between a new data point and all templates, the template with the smallest distances to the new data point was obtained, and the corresponding target was selected.

To determine the time-shift between the top-left and bottom-right position, the cross-correlation was used. As the cVEP BCI has different delays for each stimulus, the data was corrected for the corresponding delay, to obtain only the monitor raster latency.

5.1.4.4 Correcting raster latencies

As the exact position of the middle of each target row is known, their exact latency can be calculated by using the measured raster latency l_m of the used Iiyama A901HT monitor. The latency l_r of each pixel row r of the monitor can be calculated by

$$l_r = r \cdot \frac{l_m}{n} \quad (5.1)$$

where n is the number of vertical pixels depending on the resolution. Since the resolution was set to 640×480 pixels, $n = 480$ in this case. Without re-sampling of the EEG data, it can only be shifted sample-wise. Therefore, the latencies were converted and rounded to the corresponding number of samples. This allows shifting the OCSVM templates associated with a target row by the corresponding number of samples. For the 4 target rows of the presentation layout, this results in a shift of 2, 3, 5 and 6 samples, respectively. The corrected templates are used for the classification of the test trials. Of course, the trials used for spatial filter calculation were also corrected by shifting each trial.

5.1.4.5 Performance evaluation

To assess the performance difference, the target prediction accuracy was used. Because of the ceiling effect, the difference between the most probable template (smallest distance) and the second most probable template was analyzed for every single trial. An increased difference is expected by correcting the raster latencies because this indicates a clearer classification.

5.2 Results

5.2.1 Measuring raster latencies

First, the average delay and standard deviation (SD) between the trigger onsets and the real stimulus onsets at the upper left position were analyzed. For the Iiyama

monitor, this results in 0.428 ms (SD = 0.0179 ms), for the BenQ monitor in 18.39 ms (SD = 0.034 ms), and for the Dell in 4.26 ms (SD = 0.213 ms). As the standard deviations are quite low, 5 measures are enough for each measurement position. For the bottom right position, the average delay is 15.35 ms (SD = 0.0415 ms) for the Iiyama, 34.26 ms (SD = 0.137 ms) for the BenQ, and 20.14 ms (SD = 0.462 ms) for the Dell. This results in raster latency of 14.92 ms between the upper left and bottom right position for the Iiyama, 15.86 for the BenQ and 15.88 ms for the Dell. Illustrations of the measures for these monitors are depicted in Fig. 5.1.

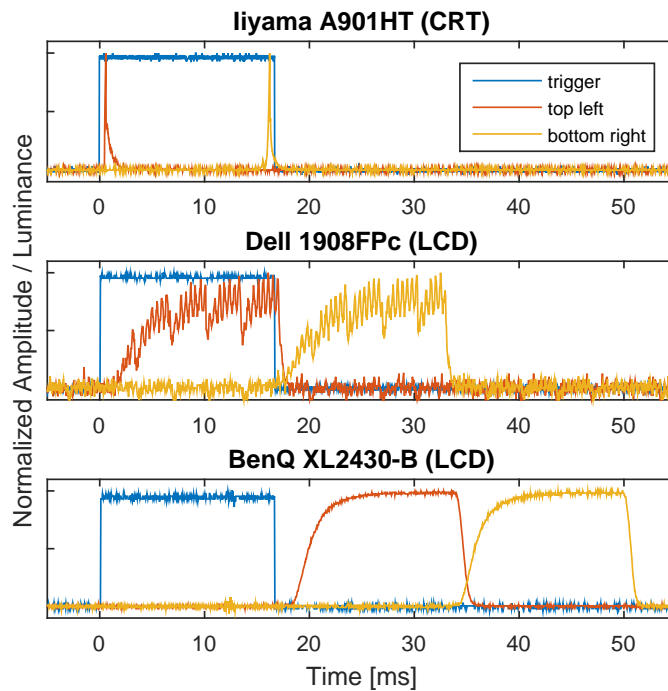


Figure 5.1: Measured raster latencies of different monitors: Iiyama A901HT, Dell 1908FPc, and BenQ XL2430-B. The blue lines indicate the triggers at which a visual stimulation was forced, for this the parallel port is used. The red and yellow lines are the measured visual stimuli of the monitor at the top left and bottom right position.

5.2.2 Effects on VEP generation

Effects on SSVEPs: The plots in Fig. 5.2 show the averaged potential of the 1-second windows of the 1 Hz and 15 Hz stimulation, respectively. The blue lines represent the top left position and the red lines the bottom right position. For the cross-correlation analysis of the 1 Hz stimulation, the first 250 ms of the VEP response was used, resulting in a shift of 9 samples which corresponds to a time-shift of 15 ms. For the 15 Hz stimulation, the full 1-second window was used for the cross-correlation analysis, also resulting in 9 samples (15 ms).

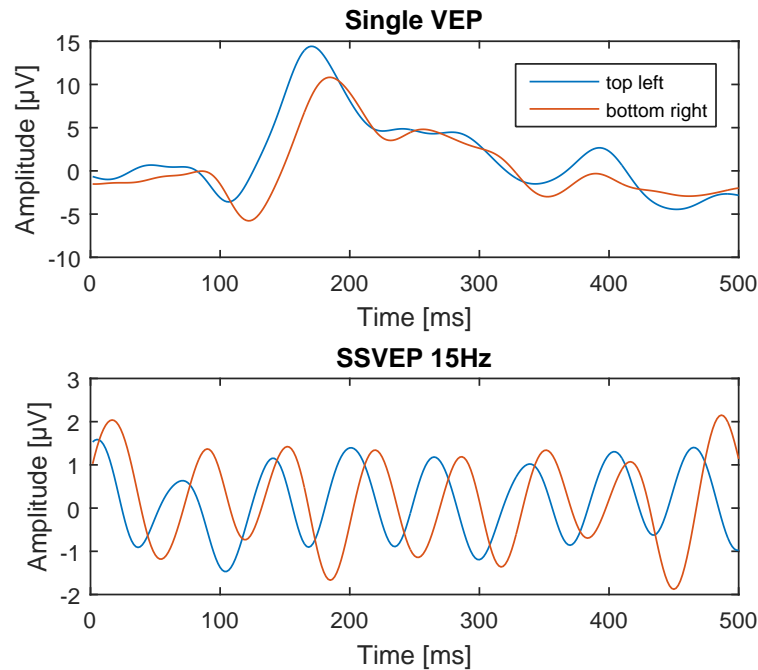


Figure 5.2: Averaged potential of the EEG data on electrode O2 collected during the SSVEP experiment using stimulation frequencies of 1 Hz (top) and 15 Hz (bottom). The blue lines correspond to the top left position and the red lines to the bottom right. Shown are windows of 500 ms.

Effects on cVEP: Analogous to the SSVEP analysis, Fig. 5.3 shows the average cVEP response of the top left and bottom right target of subject AD. The cross-correlation analysis of the averaged 1.05 s trials results in 6 samples, which corresponds to a time-shift of 10 ms.

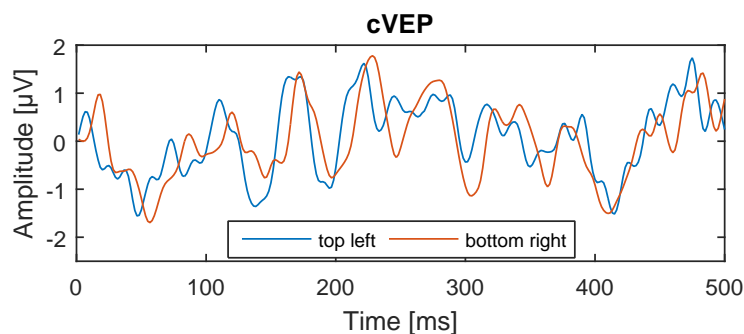


Figure 5.3: Averaged potential of the spatially filtered EEG data of the first (top left) and last (bottom right) target trials of subject AD. The blue line corresponds to the top left position and the red line to the bottom right position. Data of the bottom right target was corrected for the stimulus delay to only obtain the raster latency. Shown are the first 500 ms after trial start.

5.2.3 Effects on P300

The plot in Fig. 5.4 shows the averaged potentials of the P300 EEG data on electrode Cz for the top left and bottom right target, respectively. The cross-correlation analysis revealed a shift of 3 samples which, due to the sampling rate of 256 Hz, corresponds to a time-shift of 11.72 ms.

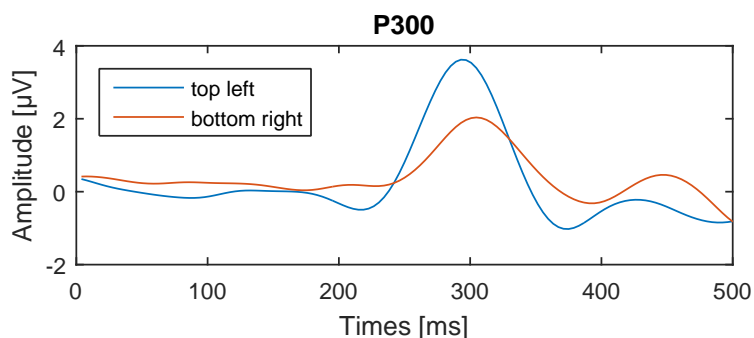


Figure 5.4: Averaged potential of the P300 EEG data on electrode Cz for the top left and the bottom right symbol in the speller matrix. The blue line corresponds to the top left position and the red line to the bottom right position. Shown are the first 500 ms after trial start.

5.2.4 Effects on BCI control performance

The previous analysis of the cVEP data reveals a time-shift of the measured cVEPs. To determine the effects on the classification of a real BCI application, the performance difference was analyzed by taking the raster latencies into account or not.

Tab. 5.1 shows the accuracies for the two sessions (S1 and S2) for each of the 9 subjects. The average accuracy of the not corrected data is 92.02% (7.98% error), whereby the accuracy of the corrected data is 95.39% (4.61% error), which in turn is a significant ($p < 0.05$, paired t-test) error reduction of 42.23%. It is also worth mentioning that correcting the data increased the performance from 60.24% to 89.41% for the subject with the lowest performance (AE S1). Furthermore, the number of perfect runs increased from 1 to 3.

Also, the difference between the Euclidean distances of the most probable and second most probable target were analyzed for all trials of all subjects, whereas only the 9326 correctly classified trials were taken into account. On average the difference is 19.03% for the not corrected data and 22.5% for the corrected data, this is a highly significant ($p < 10^{-5}$, paired t-test) increase of the difference of 18.23%.

Table 5.1: Comparison of the target prediction accuracies using raster latency correction or not.

	not corrected		corrected	
	S1	S2	S1	S2
AA	96.88%	94.62%	97.22%	97.05%
AB	80.03%	87.48%	87.84%	94.22%
AC	98.61%	98.26%	98.61%	98.26%
AD	98.96%	100.00%	98.96%	100.00%
AE	60.24%	99.31%	89.41%	100.00%
AF	97.74%	94.97%	98.09%	95.66%
AG	99.83%	98.96%	100.00%	99.31%
AH	72.74%	86.63%	78.64%	92.01%
AI	96.18%	94.97%	96.70%	94.97%
Session \emptyset	89.02%	95.02%	93.94 %	96.83%
Total \emptyset	92.02%		95.39%	
Error \emptyset	7.98%		4.61%	

Shown are the target prediction accuracies for all participant. Data acquisition was split in two sessions (S1 and S2) for nine subjects (AA to AI). Additionally, due to the ceiling effect, the target prediction error is given.

5.3 Discussion

5.3.1 Effects of raster latencies

The findings of the raster latency measures appear to be consistent with the results of Tobias Elze [Elz10]. The measured raster latency between the top left and bottom right position of 15.86 ms is nearly 95% of the refresh-cycle time of $16.\bar{6}$ ms, and it seems to be consistent for each refresh cycle and different types of monitors.

A phase-shift for SSVEPs between the top left and bottom right position was determined. As a 5 cm \times 5 cm square was used for stimulation, the vertical distance between the top and bottom square is 83.27% of the monitor's full display height, therefore, the theoretical time-shift should be 13.21 ms. The cross-correlation analysis resulted in 15 ms, but as the time resolution is bound to the sampling rate, the real shift is between $13.\bar{3}$ ms and $16.\bar{6}$ ms.

As expected, the raster latencies also affect ERPs. The analysis of the P300 data revealed a time-shift of 11.72 ms (3 samples), but due to the low sampling rate of 256 Hz, the real time-shift is between 7.81 ms and 15.63 ms. Because of the low sampling rate, accurate correction of the raster latency was not possible for each target row, therefore, the BCI target prediction performance difference was not analyzed. It is known that timings of ERPs also vary across conditions [GP02,

ICM⁺14], and Mowla *et al.* [MHT17] showed that correcting latencies in general leads to enhanced performance.

It is clearly confirmed that raster latencies have effects on both VEPs and ERPs. As a VEP BCI speller normally arranges targets over several lines of the monitor, the EEG data becomes asynchronous to the stimuli, which in turn reduces the performance. By taking the raster latency into account, the average target prediction error could be reduced from 7.98% to 4.61% for all runs of all subjects, this is a reduction of 42.23%. Since the performance, by not taking the raster latency into account, was already good the room for improvement was quite small. Because of this ceiling effect, subjects with lower initial accuracy will profit more. For example, the accuracy of the subject with the lowest performance (AE S1) has improved from 60.24% to 89.41%.

Additionally, the decision reliability was analyzed of each target classification. The reliability should be better if the distance difference between the most probable and second most probable target increases. The results reveal a highly significant increase of 18.24%, indicating a generally more reliable classification.

This clearly confirms that correcting the raster latencies is important for cVEP BCIs and will also affect all experiments depending on strict stimulus timing in a millisecond scale using a standard computer monitor for stimulus presentation, like P300, SSVEP, and other VEP experiments.

Furthermore, the average potentials of the spatially filter EEG data of every single subject of the cVEP and P300 data were analyzed and revealed an equal shift for all of them (only data for one subject is shown). This confirms that the effect of the raster latency is independent of the subject.

The effects on the performance of an SSVEP BCI depends on the classification method. A method using the frequency domain will result in the same power of frequencies, as the frequency domain is independent of the time-shift. Another time-shift independent method is the canonical-correlation analysis (CCA) with sine and cosine templates, like the method of Chen *et al.* [CWN⁺15]. For those methods, the performance difference by correcting the raster latency should be negligible. On the other side, for classification methods detecting the phase-shift [YLC⁺13, PGD⁺11] or more general which are time-dependent, the raster latency correction is highly recommended.

For BCIs which display stimuli at one constant location [JDZ⁺14], the monitor raster latency only causes a static shift from the expected onset to the real onset, which should not affect the performance a lot.

5.3.2 Addressing raster latencies

Besides the used method for correcting the raster latencies for a cVEP BCI by time-shifting the templates, it is also possible to use a monitor with a higher refresh rate to reduce the raster latencies. New monitors reach refresh rates up to 240 Hz and quadrupling the refresh rate means the delay between the first and last line is divided by four which is a maximum raster latency of $4.1\bar{6}$ ms.

Alternatively, as a time-shift only occurs if the stimuli emitting unit causes raster latencies, one can use other stimuli emitting units, like LEDs [HLJ⁺12], indeed this depends on the experiment itself.

Furthermore, one can use classification methods which are not affected by the time-shift, like CCA or using only the frequency domain in case of SSVEPs, as mentioned above.

Chapter 6

EEG2Code: predicting stimulation patterns

The general understanding of Brain-Computer Interface (BCI) is, that it enables users to actively control a computer by using brain activity. Those BCIs are allowing active control, are called *active* or *explicit* BCIs. Contrary, there are *passive* BCIs also called *implicit*, which do not allow active control, but to the brain activity is used, for example, to adapt the application to the user's mental state. Passive BCIs are mostly used for workload detection [GSF⁺17, WRB⁺17, KSR⁺18] or emotion detection [MHH17, AC16]. In general, an implicit interaction can be defined as "an action performed by the user that is not primarily aimed to interact with a computerized system but which such a system understands as input" [Sch00].

Most of the active BCIs are based on event-related potentials (ERPs) or visual evoked potentials (VEPs). The latter are brain responses to flickering visual stimuli. VEP BCIs are either based on the original idea of Sutter [Sut84] and use complex stimulation patterns (also called codes) to elicit a code-modulated VEP (cVEP) [SRB12b, BGW⁺11], or they are based on visual stimuli modulated by specific frequencies which evoke steady-state VEPs (SSVEPs) [CWN⁺15]. For both methods, the number of possible stimulation patterns optimized for a specific target layout. For example, the SSVEP BCI speller by Chen *et al.* [CWN⁺15] was designed for a 40-target layout and adding more targets would require to redesign the stimuli completely.

In this chapter, a new stimulation paradigm is proposed, which does not design a limited set of stimulation patterns for a specific purpose but make use of fully random stimulation patterns. Furthermore, a model is presented which can predict stimulation patterns of arbitrary VEPs. Therefore, it is an entirely new approach implementing a passive BCI which allows extracting visual information that is contained in the EEG. But it should be noted, that the usability of stimulation patterns is restricted by the brain. Herrmann [Her01] has shown that brain responses can only be

found by using stimulation frequencies of up to 90 Hz. Furthermore, the stimulation rate is limited by the used hardware for stimulus presentation, like a computer monitor. Therefore, "arbitrary stimulation patterns" should be interpreted as a huge set of possible stimulation patterns relative to the used stimulation rate.

The general idea of the EEG2Code model was published in [NRS17], whereas parts of the results, figures and wording are already published in [NS18].

6.1 Methods

6.1.1 Training

The model is based on a ridge regression model, which uses the EEG signal to predict the stimulation pattern during an arbitrary stimulation. The current work is based on flash VEPs and considering their waveform (see section 2.3.1), it can be assumed that the most prominent waves last for around 250 ms to 300 ms. Therefore, a 250 ms window of spatially filtered EEG data is used as predictor and the corresponding bit of the stimulation pattern (0=black, 1=white) as response to train the ridge regression model. The window is shifted sample-wise over the data during a trial, meaning that it is required to use 250 ms of EEG data after trial-end, otherwise, the last 250 ms of a stimulation pattern cannot be predicted. Fig. 6.1(A) depicts this procedure with a bit-wise window shifting for simplicity. The ridge regression model is trained using Eq. 2.2 and Eq. 2.3, where X is a $n \times k$ -matrix with n the number of windows and k the window length (number of samples). y is a $n \times 1$ -vector containing the corresponding bit of the stimulation sequence for each window. The

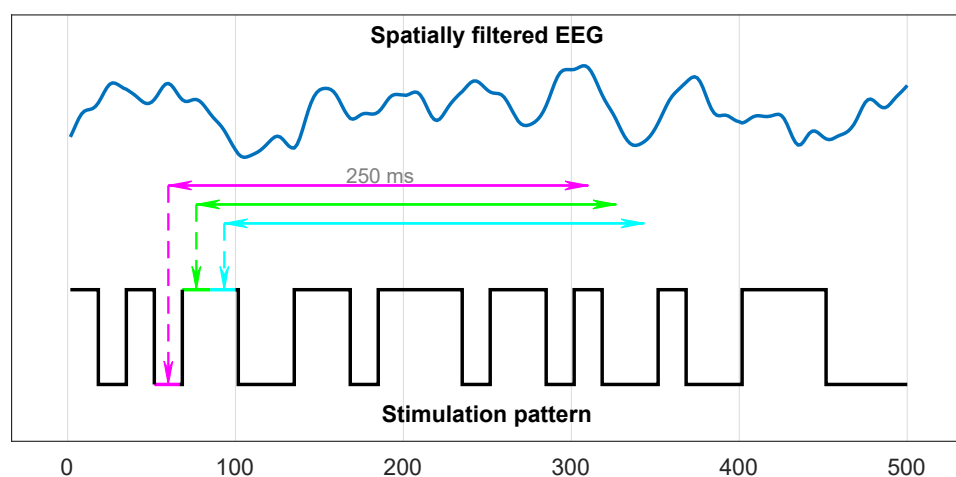


Figure 6.1: Training of the EEG2Code regression model. Each 250ms window of the spatially filtered EEG data will be projected to its corresponding bit (1 or 0) of the corresponding stimulation pattern.

regularization parameter λ was not optimized but set to 0.001. Since a window has a length of $k = 150$ samples, at the used sampling rate $s = 600\text{Hz}$, the output $\hat{\beta}$ is a coefficient vector of length 150, one for each input sample and the constant bias term β_0 . The number of windows n depends on the number of trials N , the average trial duration d , the window length k , and the sampling rate s :

$$n = N(d \cdot s - k) \quad (6.1)$$

As described in section 6.1.5, $N = 96$ and $d = 4s$ were used resulting in $n = 216,000$ windows.

6.1.2 Prediction

After training, the model is able to predict the bits of the stimulation pattern. Fig. 6.2 depicts the procedure, in which the measured EEG data is spatially filtered, and the trained regression model is used to predict each 250 ms window (sample-wise shifted). The regression model predicts a real number y_i for each window i , whereby y is calculated using Eq. 2.4. The predicted real values y_i , in turn, can be interpreted as binary values by a simple threshold method, each value above or equal 0.5 is set to be binary 1 and 0 otherwise. Afterward, the predicted binary sequence is compared to the stimulation pattern using the Hamming distance.

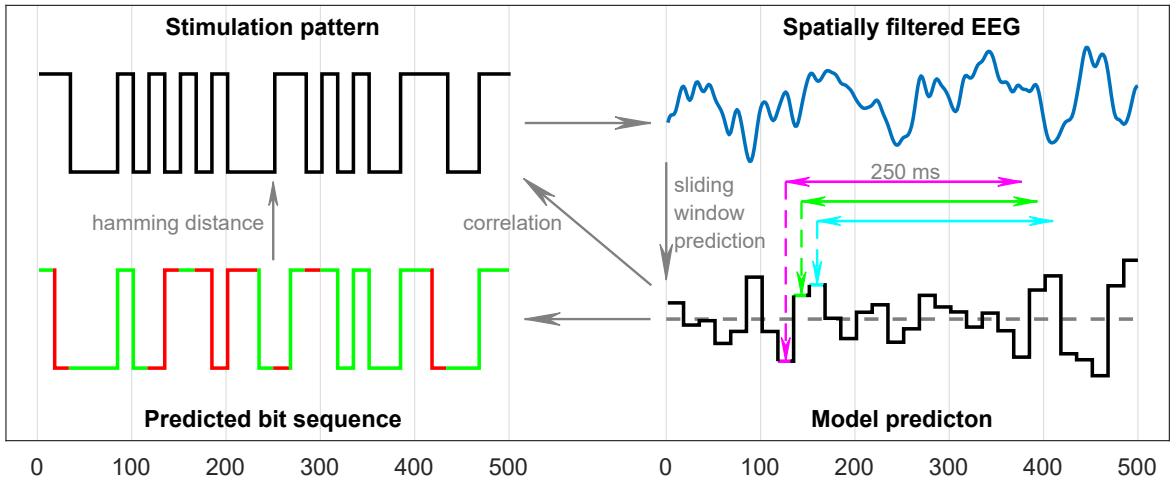


Figure 6.2: The EEG2Code model predicts an arbitrary stimulation pattern. A 250ms window will be slid sample-wise over the spatially filtered EEG signal. For simplicity, it is shown bit-wise using 3 exemplary windows. The trained model predicts a real number for each window. Each value above 0.5 (gray dashed line) is interpreted as boolean 1 or 0, otherwise. The resulting bit sequence can be compared (Hamming distance) to the stimulation pattern (match = green, mismatch = red).

6.1.3 Random modulation patterns

During the experiment, random stimulation patterns are used (see section 4.3). At each monitor refresh a random integer (0 or 1) is generated for each target of the used presentation layout, therefore, the binary sequence of a target is always random without conscious repetitions and generated with a rate of 60 bit/s, continuously. The pattern generation can be repeated or varied by using the same or a different random seed, respectively. Fig. 6.3 shows four exemplary stimulation patterns.

6.1.4 Hardware & Software

The experimental setup is as shown in Fig. 6.3. The system consists of a g.USBamp (g.tec, Austria) EEG amplifier, two personal computers (PCs), Brainproducts Acticap system with 32 channels and an LCD monitor (BenQ XL2430-B) for stimuli presentation. Participants are seated approximately 80 cm in front of the monitor.

PC1 is used for the presentation on the LCD monitor, which is set to refresh rate of 60 Hz and its native resolution of 1920×1080 pixels. The stimuli are presented using the framework introduced in chapter 4. A stimulus can either be black or white, which can be represented by 0 or 1 in a binary sequence and is synchronized with the refresh rate of the LCD monitor. The timings of the monitor refresh cycles are synchronized with the EEG amplifier by using the parallel port.

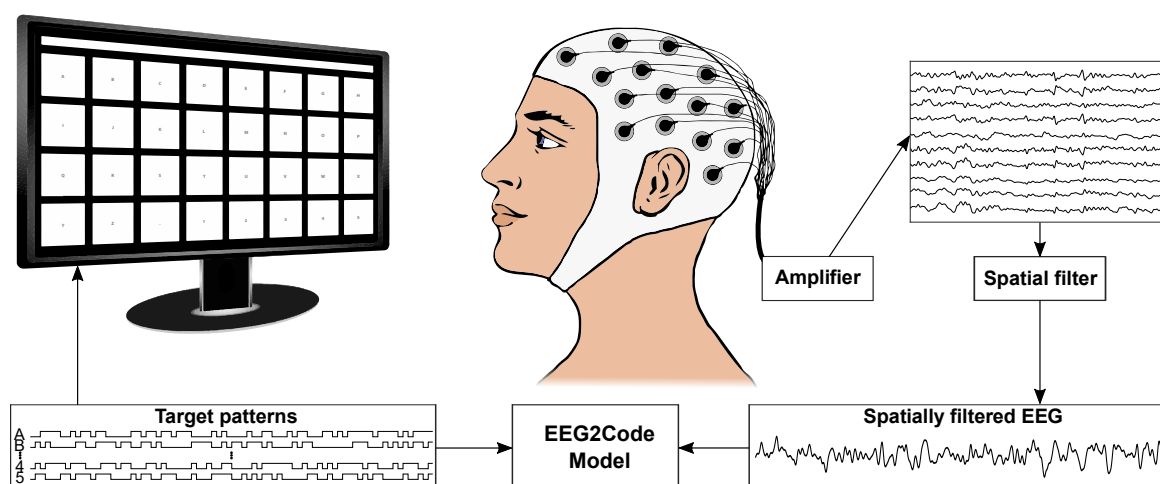


Figure 6.3: Setup of the experiment. The presentation layout is as shown on the monitor, it has 32 targets which are separated by a blank black space. Each target is modulated with its own random stimulation pattern. During a trial, the participant has to focus a target. The measured EEG data is amplified, and a spatial filter is applied. Afterwards, the EEGCode model uses the spatially filtered EEG and predicts the stimulation pattern as shown in Fig. 6.2.

PC2 is used for data acquisition and analysis. BCI2000 [SMH⁺04] is used as a general framework for recording the data of the EEG amplifier, and the data processing is done with MATLAB [MAT17]. The amplifier sampling rate was set to 600 Hz, resulting in 10 samples per frame/stimulus. Additionally, a TCP network connection was established to PC1 to send instructions to the presentation layer and to get the modulation patterns of the presented stimuli.

A 32 electrodes EEG layout was used, 30 electrodes were located at Fz, T7, C3, Cz, C4, T8, CP3, CPz, CP4, P5, P3, P1, Pz, P2, P4, P6, PO9, PO7, PO3, POz, PO4, PO8, PO10, O1, POO1, POO2, O2, OI1h, OI2h, and Iz. The remaining two electrodes were used for electrooculography (EOG), one between the eyes and one left of the left eye. The ground electrode (GND) was positioned at FCz and reference electrode (REF) at OZ.

6.1.5 Data acquisition

To test the model, 9 healthy subjects (5 female) were recruited. All subjects had normal or corrected-to-normal vision. The age ranged from 18 to 23 years. Each subject participated in one session and completed the whole experiment. None of the subjects participated in other VEP EEG studies before.

Initially, the participants had to perform 2 runs to generate a spatial filter (see section 6.1.6.3). The training phase was split into 3 runs, but with varying trial duration, 5 s, 4 s, and 3 s, respectively. The testing phase consists of 7 runs with a trial duration of 2 s.

During all runs, the inter-trial time was set to 0.75 s, and the participants had to perform 32 trials in lexicographic order using the matrix keyboard layout (see section 4.4.1). This means that $3 \cdot 32 \cdot 4 \cdot 60 = 23040$ random stimuli were presented during the training trials and $7 \cdot 32 \cdot 2 \cdot 60 = 26880$ stimuli during the testing trials.

6.1.6 Preprocessing

Several preprocessing steps were performed prior to the training as well as prior to the prediction of the EEG2Code model. They will be explained in the following subsections.

6.1.6.1 Frequency filter

The recorded EEG data is bandpass filtered by the amplifier between 0.1 Hz and 60 Hz using a Chebyshev filter of order 8. Furthermore, the amplifier was also configured with a notch-filter at 50 Hz to suppress the 50-Hz power line interference.

6.1.6.2 Correcting raster latencies

As described in chapter 5, standard computer monitors (CRT, LCD) cause raster latencies because of the line by line image build-up dependent on the refresh rate. As the targets of the matrix-keyboard layout are arranged in multiple rows, the raster latency was corrected as described in section 5.1.4.4.

For the used presentation layout (Fig. 4.1), the vertical center of the four target rows r are at 231, 463, 695, and 927 pixels. Using Eq. 5.1 and the used vertical resolution of $n = 1080$ pixels and the measured monitor raster latency $l_m = 15.86$ ms (see section 5.2.1), the latencies l_r for each row are 3.39 ms, 6.80 ms, 10.21 ms, and 13.61 ms. With the used sampling rate of 600 Hz of the EEG amplifier, the monitor raster latency is corrected by shifting the EEG data by $\text{round}(l_r \cdot 600/1000)$ samples, which results in 2, 4, 6, or 8 samples, respectively.

6.1.6.3 Spatial filter

Recent studies [BGW⁺11, SRB12a] have shown increased classification accuracy by using spatial filters to improve the signal-to-noise ratio of the brain signals. As random stimulation is not suitable for spatial filter training, an m-sequence with low auto-correlation is used for target modulation. The spatial filter training is done using a canonical correlation analysis (CCA) as described in [SWRB14], except that the presentation layout and stimulation duration differ. The presentation layout is as described above, and the participants had to perform 32 trials (one per target) whereby one trial lasts for 3.15 seconds followed by 1.05 for gaze shifting. As the used modulation pattern has a length of 63 bits (1.05 seconds), this results in 96 sequence cycles per participant, which in turn are used for training the spatial filter. The spatial filter is then used for the following experiment.

6.1.7 Performance evaluation

For the pattern prediction of the EEG2Code model, the binary values of the predicted patterns were compared to the stimulation patterns by calculating the accuracy using Eq. 2.9, which corresponds to how much of the stimulation pattern can be predicted correctly.

In addition to the accuracies, the corresponding ITRs were calculated using Eq. 2.10 with $N = 2$ (1 or 0), P the corresponding pattern prediction accuracy, and $T = 1/60$ as a single stimulus lasts for 1/60 s, and each bit was predicted individually.

6.2 Results

The average results for each participant are shown in Table 6.1. When using random visual stimulation, the EEG2Code model is able to predict the random stimulation pattern with an average accuracy of 64.6% (the percentage of correctly predicted bits in the bit-vector), which corresponds to an ITR of 232 bit per minute (bpm). For the best subject, the EEG2Code model can predict 69.1% of the random stimulation patterns correctly, which corresponds to an ITR of 389.9 bpm.

Additionally, the spatial filter trials were used to predict the corresponding m-sequence. In average, the bit-prediction results in 65.9%, which corresponds to an average ITR of 275.0 bpm. It is worth to note that this ITR is just for comparison because the cardinality of random stimulation patterns is higher as the cardinality of a shifted m-sequence.

Table 6.1: Results of the EEG2Code model prediction.

Subject	Random stimulation		m-sequence	
	ACC [%]	ITR [bpm]	ACC [%]	ITR [bpm]
S1	69.1	389.9	69.7	414.4
S2	64.5	222.4	65.9	267.7
S3	63.7	196.5	64.9	235.3
S4	65.6	257.8	70.0	429.4
S5	66.3	282.7	65.7	260.5
S6	67.1	308.2	66.8	300.5
S7	63.7	196.8	64.6	225.3
S8	60.9	124.5	62.9	173.5
S9	60.2	109.4	62.7	168.7
mean	64.6	232.0	65.9	275.0

Shown are the average results of all subjects, whereby the best results are in bold font. For the EEG2Code model, the accuracy (ACC) indicates how many bits of the stimulation pattern can be predicted correctly. The ITR is calculated using Eq. 2.10 with $N = 2$ and $T = 1/60s$. For the m-sequence prediction, the ITRs are just for comparison.

6.3 Discussion

Common approaches are based on predefined stimulation patterns, like specific frequencies (SSVEP) or patterns with low auto-correlation (cVEP), therefore, they are based on a limited set of possible stimulation patterns. Contrary, the EEG2Code model, presented in this chapter, is a novel approach. It is the first method which allows predicting arbitrary stimulation patterns, therefore, it offers new possibilities. First and foremost, the set of possible stimulation patterns is only limited by

the presentation speed. In the current work, the stimuli were presented with a rate of 60 bit/s which in turn allows having a set of $2^{60} = 1.15 \cdot 10^{18}$ possible stimulation patterns when using 1 s of EEG data.

The results have shown that the fully random stimulation patterns can be predicted with an average accuracy of 64.6 %, which corresponds to an ITR of 232 bpm (with a maximum of 69.1 % and 389.9 bpm). It must be noted that for the ITR calculation $T = 1/60s$ was used which is slightly biased due to the required window length of 250 ms, but T approaches to $1/60s$ and the bias decreases with increasing trial duration. Furthermore, these ITRs reflect the visual information that can be extracted from the EEG, but it cannot be compared to ITRs during active BCI control, because the user cannot interact. It must be seen as a method which allows transferring information (visual stimuli) with the human brain as a data channel.

If the accuracy could be further increased, for example by presenting the same stimulation pattern multiple times, it would be possible to use error-correction codes, how they are used for common data transfer methods.

Interestingly, the bit-prediction accuracies of the m-sequence are significantly ($p < 0.05$, paired t-test) higher as the prediction of random stimulation patterns. This will be further addressed in chapter 8. In chapter 9 it will be shown how the model can be used for active BCI control.

Finally, the approach could also be applied to other stimulation paradigms like sensory or auditory stimulation.

Chapter 7

Code2EEG: predicting brain responses to visual stimuli

As explained, the EEG2Code model presented in the previous chapter, is a novel approach, as it predicts arbitrary stimulation patterns. Contrary, most of the recent approaches [SRB12a,CWN⁺15] construct/train templates of EEG data for each used stimulation pattern. The templates, in turn, are used to classify a new trial by comparing the new EEG data to each template.

One notable exception was proposed by Thielen *et al.* [TvdBFD15]. They developed a method which is able to predict the brain responses to compositions of short and long stimulation pulses, which allows creating multiple templates without a specific training of the composition. While this approach allows for a more flexible prediction model, the stimulation patterns are not fully arbitrary, as they can only be composed of short and long pulses. Again, "arbitrary stimulation patterns" should be interpreted as a huge set of possible stimulation patterns relative to the used stimulation rate.

In this chapter, the backward model of the EEG2Code model is presented, called Code2EEG. The model is able to predict the brains response to arbitrary stimulation patterns, therefore, it can be used to create templates for each possible stimulation pattern without a specific training of the stimulation pattern. Furthermore, it will be shown that the predicted brain responses highly correlate with the measured EEG using the same stimulation pattern, even on a single trial basis.

Parts of the results, figures and wording are already published in [NS18].

7.1 Methods

7.1.1 Training

Like the EEG2Code model, the Code2EEG model is based on a ridge regression model and trained on fully random stimulation patterns to predict the EEG response based on a stimulation pattern. Also, a 250 ms windows (150 samples) is used and the model is calculated using Eq. 2.2 and 2.3, where X is the window containing the stimulation pattern and y is the spatially filtered EEG data for the corresponding time-point. λ was set to 0.001. The output $\hat{\beta}$ of the ridge regression is a coefficient vector of length 150. Fig. 7.1 depicts this procedure for three exemplary windows.

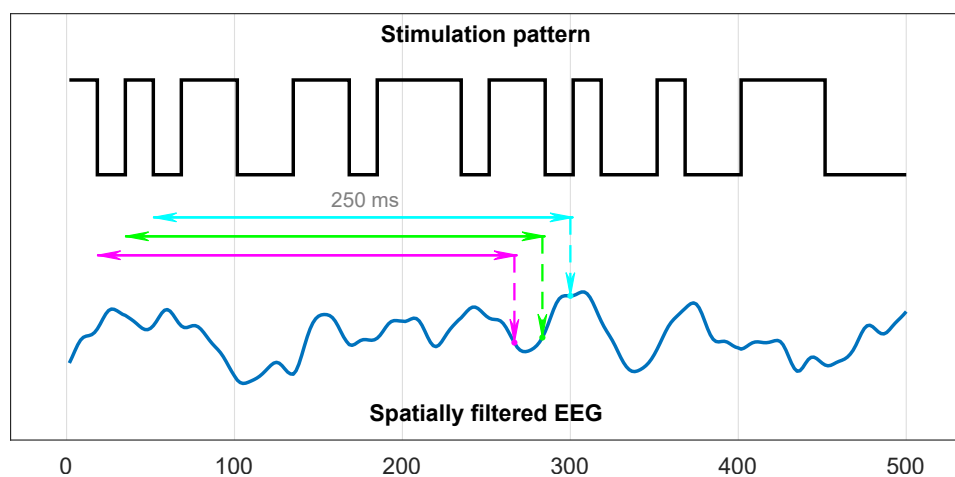


Figure 7.1: Training of the Code2EEG regression model. Each 250ms window of the modulation pattern will be projected to the corresponding value of the spatially filtered EEG data.

7.1.2 Prediction

After training, the Code2EEG model is able to predict the brain response to an arbitrary stimulation pattern. Eq. 2.4 is used for prediction, where $x_1 \dots x_k$ are the k samples of the stimulation pattern window. The model prediction y can now be compared to the measured (and spatially filtered) EEG. Fig. 7.2 depicts the procedure.

7.1.3 Hardware & Software

The same setup was used as explained in section 6.1.4. For the sake of completeness, it will be explained again.

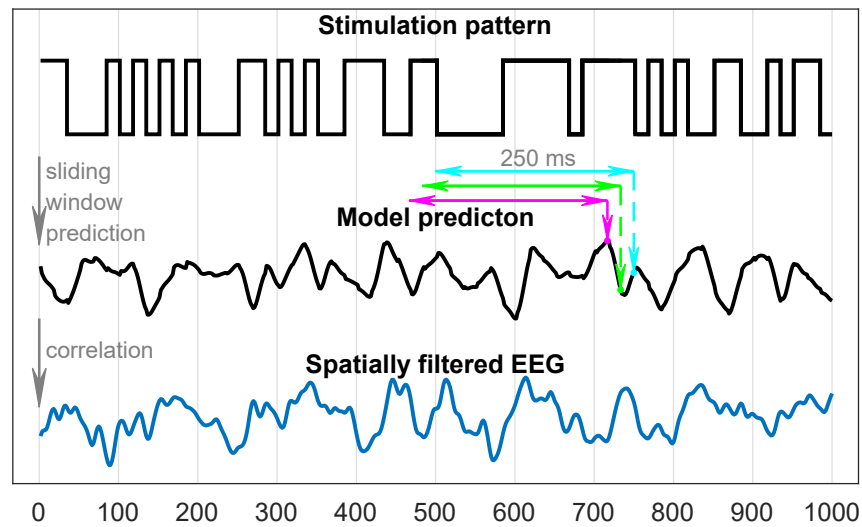


Figure 7.2: The Code2EEG model predicts the brains response to an arbitrary stimulation pattern by sliding the 250 ms window over the stimulation pattern. For simplicity, it is shown bit-wise using 3 exemplary windows.

The system consists of a g.USBamp (g.tec, Austria) EEG amplifier, two personal computers (PCs), Brainproducts Acticap system with 32 channels and an LCD monitor (BenQ XL2430-B) for stimuli presentation. Participants are seated approximately 80 cm in front of the monitor.

PC1 is used for the presentation on the LCD monitor, which is set to refresh rate of 60 Hz and its native resolution of 1920×1080 pixels. The stimuli are presented using the framework introduced in chapter 4. A stimulus can either be black or white, which can be represented by 0 or 1 in a binary sequence and is synchronized with the refresh rate of the LCD monitor. The timings of the monitor refresh cycles are synchronized with the EEG amplifier by using the parallel port.

PC2 is used for data acquisition and analysis. BCI2000 [SMH⁺04] is used as a general framework for recording the data of the EEG amplifier, and the data processing is done with MATLAB [MAT17]. The amplifier sampling rate was set to 600 Hz, resulting in 10 samples per frame/stimulus. Additionally, a TCP network connection was established to PC1 to send instructions to the presentation layer and to get the modulation patterns of the presented stimuli.

A 32 electrodes EEG layout was used, 30 electrodes were located at Fz, T7, C3, Cz, C4, T8, CP3, CPz, CP4, P5, P3, P1, Pz, P2, P4, P6, PO9, PO7, PO3, POz, PO4, PO8, PO10, O1, POO1, POO2, O2, OI1h, OI2h, and Iz. The remaining two electrodes were used for electrooculography (EOG), one between the eyes and one left of the left eye. The ground electrode (GND) was positioned at FCz and reference electrode (REF) at OZ.

7.1.4 Data acquisition

The data from the previous experiment was used, as described in section 6.1.5. Also, the same preprocessing steps were performed, as described in section 6.1.6. For the sake of completeness, the important parts will be explained again.

Initially, the participants had to perform 2 runs to generate a spatial filter (see section 6.1.6.3). The training phase was split into 3 runs, but with varying trial duration, 5 s, 4 s, and 3 s, respectively. The testing phase consists of 7 runs with a trial duration of 2 s.

During all runs, the inter-trial time was set to 0.75 s, and the participants had to perform 32 trials in lexicographic order using the matrix keyboard layout (see section 4.4.1), whereby the targets were modulated with fully random stimulation patterns (see section 4.3).

Furthermore, as the Code2EEG model can predict the brain response to an arbitrary stimulation pattern, it was tested with SSVEP and cVEP stimulation patterns. For this, an additional participant was recruited and had to perform the spatial filter and training session as described. Additionally, the participant had to perform 120 single flash trials (1 Hz) and 120 SSVEP trials using 30 Hz stimulation frequency, whereby each trial lasts for 1 s.

Both the 1 Hz and the 30 Hz pattern were presented using a single target in the middle of the screen, whereby it had the same size as the targets of the matrix keyboard layout (Fig. 4.1). The m-sequences were presented as part of the data acquisition used for spatial filter generation (see section 6.1.6.3).

7.1.5 Performance evaluation

Contrary to the EEG2Code model prediction, it is not possible to calculate the accuracy. Therefore, the predictions of the brain responses are evaluated by using the correlation coefficient between the predicted EEG and the spatially filtered EEG. Additionally, to obtain a significance measure, the p-values are calculated under the hypothesis that the correlation between the model prediction and the stimulation pattern are equal to zero.

7.2 Results

To evaluate the Code2EEG model, for each trial the stimulation pattern is used to predict the corresponding brain response. The predicted response is then compared to the measured spatially filtered EEG on a single trial basis, using the Pearson correlation coefficient. This resulted in an average of $r = 0.384$ using fully random stimulation patterns. The maximum is reached for subject S1 with an average of $r = 0.509$, which corresponds to a coefficient of determination of $r^2 = 0.259$. Additionally, the p-values, under the hypothesis that the correlation is equal to zero are given, whereby an average p-value of $1.3e^{-03}$ (up to $2.8e^{-17}$) was achieved. As the p-values take the length of the trials into account, they prove that the EEG2Code model prediction significantly correlates with the measured EEG response. Detailed results for each participant are listed in Table 7.1.

Table 7.1: Results of the Code2EEG model prediction.

Subject	r	P_t
S1	0.509	$2.8e^{-17}$
S2	0.338	$2.7e^{-04}$
S3	0.364	$8.0e^{-03}$
S4	0.396	$1.1e^{-04}$
S5	0.464	$8.9e^{-16}$
S6	0.409	$8.5e^{-08}$
S7	0.270	$4.4e^{-08}$
S8	0.344	$9.1e^{-06}$
S9	0.257	$3.2e^{-03}$
mean	0.384	$1.3e^{-03}$

Shown are the average results of all subjects, whereby best results are in bold font. The correlation (r) between the model prediction the measured EEG data is given as well as the p-values P under the hypothesis that the correlation between the model prediction and the stimulation pattern are equal to zero.

Beside using random stimulation patterns, the Code2EEG model was also used to predict the response to a single-light flash of 16.6 ms, to a 30 Hz SSVEP pattern, and to a cVEP m-sequence pattern of a single participant. The results are depicted in Fig. 7.3, whereby each plot shows the predicted brain responses of the Code2EEG model compared to the measured EEG averaged over multiple trials. It can clearly be seen, that most of the VEP responses can be predicted, except the on-set.

To address this, the train data was filtered by stimulation patterns with fewer bit changes (< 3 within 250 ms) to get mainly on-set responses. Using this data, an additional Code2EEG model was trained, and the results show that the onset for all tested stimulation patterns matches better, whereas the further prediction matches worse (Fig. 7.4)

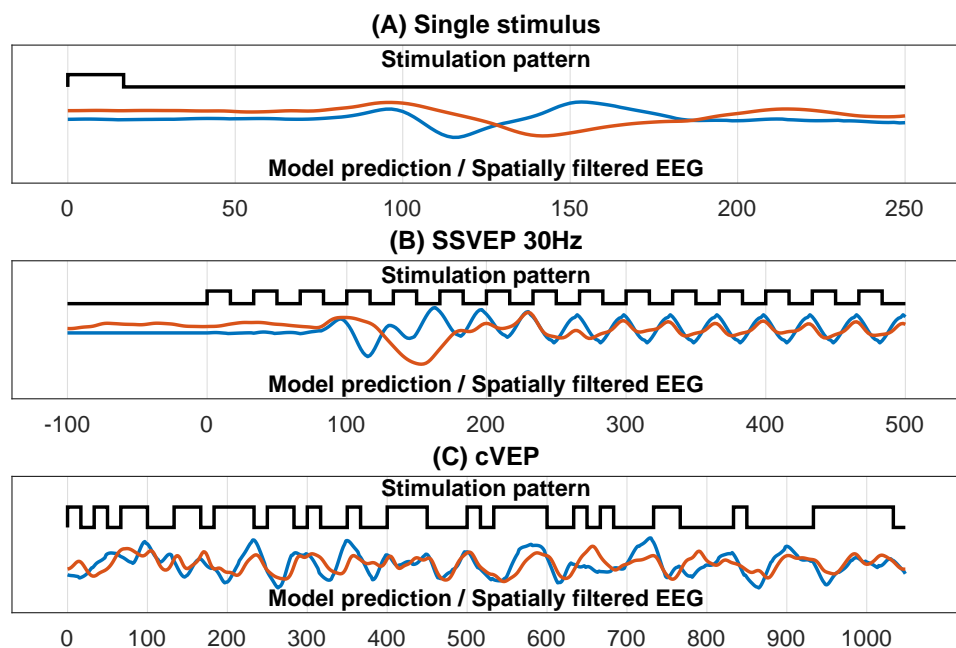


Figure 7.3: Predicted brain responses of the Code2EEG model compared to the measured EEG. The black line represents the stimulation pattern, the blue line the predicted brain response and the red line the spatially filtered EEG (120 trials averaged).

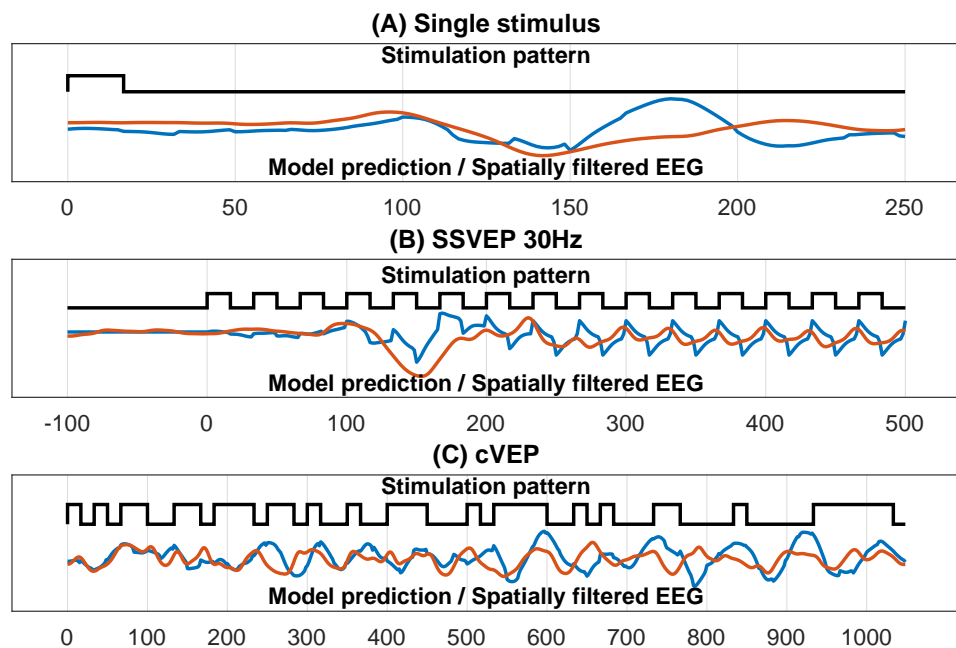


Figure 7.4: Predicted brain responses of the Code2EEG model compared to the measured EEG. For the training, only stimulation patterns with < 3 bit-changes are used. The black line represents the stimulation pattern, the blue line the predicted brain response and the red line the spatially filtered EEG (120 trials averaged).

7.3 Discussion

In this chapter, a model was presented which is able to predict the brain responses to visual stimulation patterns. Contrary to comparable methods [TvdBFD15, CCA⁺16], the presented method works with arbitrary stimulation patterns, while it is only trained on a limited set of stimulation patterns. Random stimulation patterns were used because it was assumed to cover most of the possible VEP responses.

Using fully random stimulation patterns, the correlation between the recorded EEG and the predicted spatially-filtered EEG response results in $r = 0.384$. It should be noted that the Code2EEG model only predicts the evoked response in the EEG, and not the noise present in the EEG so that the correlation between the prediction and the evoked response should actually be higher. On the other side, a correlation of $r = 0.384$ means that $r^2 = 14.75\%$ of the variance in the recorded EEG can be explained by the Code2EEG model, thereby giving a lower bound for the signal-to-noise-ratio (SNR) of the visual evoked response in the EEG. This is confirmed by comparing the model prediction of the cVEP pattern to the averaged (and thereby noise reduced) recorded EEG (Fig.7.3(C)), which resulted in a correlation of $r = 0.551$ ($r^2 = 30.4\%$).

Interestingly participant S8 achieved a similar correlation as S3 (0.344 vs. 0.364), but the bit-prediction accuracy of S3 is clearly increased (63.7 bit/min vs. 60.9 bit/min).

The only comparable work is by Thielen *et al.* [TvdBFD15]. But it must be noted that, although the method could be used with any stimulation pattern composed of short and long pulses, they constructed two optimized distinct sets containing 65 pre-constructed 126-bit long stimulation patterns each, one for training and one for testing, respectively. Unfortunately, although they used the method for BCI control, they did not analyze the brain response prediction of their model on a single trial basis but averaged over multiple trials, which resulted in $r^2 = 47.6\%$. At first glance, this suggests a better performance, but as noted, the prediction is compared to an averaged EEG, furthermore, only 65 stimulation patterns were tested. Contrary, the presented model in this work was tested on fully random stimulation patterns without conscious repetitions, which is why the responses cannot be averaged.

A further approach predicting SSVEPs is by Wang *et al.* [WGG15]. They constructed a computational model which is able to predict SSVEPs for a given frequency. Unfortunately, the model was only evaluated for BCI control. Furthermore, an SSVEP response shows a (more or less) static course, as shown in Fig. 7.3(b), which is obviously easier to predict as complex VEPs, like cVEPs (see Fig. 7.3(c)).

Another approach predicting VEPs was recently published by Sobreira *et al.* [STK18], but it is not directly comparable. They constructed a deep-learning model which simulates the visual pathway. Their model is able to predict VEPs to a single stimulus based on the shape of the stimulus, which resulted in a median

correlation of $r = 0.91$. But it must be emphasized that the method is not able to predict any other (complex) VEPs, which means it is not suitable for BCI control.

Finally, the presented approach is based on the assumption of linearity in the VEP generation process [CPAD⁺11, LPR⁺06], it was shown that most of the VEP response to arbitrary stimulation patterns can be explained by a superposition of single VEP responses. But interestingly, the duration of the predicted single-flash VEP response and the onset of the predicted 30 Hz SSVEP response is shortened in time compared to the recorded VEP responses, whereas the sinusoidal part of the 30 Hz SSVEP response matches well. It seems there is an initially slowed response after a longer stimulation pause which the model cannot reflect because it is trained using random stimulation patterns where longer stimulation pauses are unlikely. To address this, a model was trained using only data with fewer bit changes (< 3 within 250 ms). The results show that the on-set for all evaluated stimulation patterns matches good, whereas the further prediction matches worse (Fig. 7.4). This could be an evidence, that the VEP generation is a non-linear process and will be further addressed in chapter 13 by using a non-linear approach.

In chapter 9 it will be shown how the Code2EEG model can be used for BCI control.

Chapter 8

Optimization of stimulation patterns

As explained in section 2.3.1, the most prominent parts of a flash VEP are P2, N3, and P3, the negative/positive potentials with peaks at around 110 ms, 150 ms, and 220 ms, (post-stimulus), respectively. But this means that VEPs will mix-up for successive stimuli with a time interval lower than approximately 250 ms. It is not entirely clear how this mix-up proceeds, as it might be a simple overlap of responses or the brain might be entrained to the modulated stimuli [Her01].

Obviously, not each possible stimulation pattern could result in equal performances. Imagine, a stimulation pattern which is always off or on would not result in a VEP, as VEPs are evoked because of flickering. Furthermore, Herrmann [Her01] has shown that VEPs can only be found by using stimulation frequencies of up to 90 Hz, which is most probably caused due to the refractory period of neurons. Therefore, to get the best classification performances, it is required to find modulation patterns evoking brain responses which can be differentiated between others as effectively as possible.

In chapter 6 it was found that the bit-prediction accuracy using m-sequence stimulation is significantly better as the random bit-prediction accuracy. Since the EEG2Code model allows to use arbitrary stimulation patterns, it is possible to search for specific properties of stimulation patterns which results in better classification performances. In this chapter a specific property was tested which resulted in an increased performance.

The general idea of the pattern optimization was published in [NRS18], whereas parts of the results, figures and wording are already published in [NS18].

8.1 Methods

8.1.1 Analyzing the m-sequence

As explained, not each possible stimulation pattern could result in equal performances. Since the bit-prediction accuracies of the m-sequence are significantly better as of random stimulation patterns, there must be a specific property of the m-sequence.

The prediction of the EEG2Code model takes only 250 ms windows of EEG data into account, therefore, the low auto-correlation of the m-sequence cannot be responsible for the increased performance. As mentioned, a stimulation pattern which is always off or on would not result in a VEP, therefore, a specific property could be the number of bit-changes. Again, since the EEG2Code model takes only 250 ms (15-bit) windows into account, all 15-bit long sub-sequences of the m-sequence were analyzed for the number of bit-changes. To clarify, a bit-change is the change of binary 1 to 0 and vice versa, respectively.

8.1.2 Effects of the number of bit-changes

To prove if the findings of the m-sequence analysis, also the bit-prediction accuracies using random stimulation patterns were analyzed. Using the data of chapter 6, an offline analysis was performed by splitting each trial of the 7 test runs into all possible 250 ms long sub-trials. For each of those sub-trials, the bit-prediction accuracy and the corresponding number of bit-changes were determined. Since the number of bit-changes depends on the length of the bit sequence, the bit-change probability will be given, which is the percentage of possible bit-changes. For example, a bit-change probability of 100% means that each successive bit-changes from 0 to 1 or 1 to 0, respectively.

8.1.3 Generate optimized stimulation patterns

Based on the findings of the bit-change analysis (see results), a set of all 15-bit long sequences with 50% bit-change probability (7 bit-changes) was generated. This results in a total number of 6,864 15-bit sequences.

In preparation of the BCI control, where different stimulation patterns should evoke VEPs that are as different as possible, the set of the 6,864 sequences is further reduced, to get a set of sequences that are as dissimilar to each other as possible.

For this, 100,000 sets of 150 randomly chosen sequences out of the 6,864 bit-sequences were generated. The set with the lowest average correlation coefficient

between each sequence to all others was selected. The resultant subset has an average correlation of -0.004 ($SD = 0.276$) between any sub-sequence to all others.

Since a 32-target matrix keyboard layout was used, the 15-bit sequences of the optimized set must be assigned to the targets. This is done randomly without repetitions, which means that the same sequence will only be assigned to a single target. Once the sequence is presented, which took 250 ms using a presentation rate of 60 bit/sec, a new one will be assigned. Contrary to assigning a static sequence to each target, this allows modulating $150^{T/250\text{ms}}$ different targets, where T denotes the trial duration in milliseconds.

8.1.4 Hardware & Software

The same setup was used as explained in section 6.1.4. For the sake of completeness, it will be explained again.

The system consists of a g.USBamp (g.tec, Austria) EEG amplifier, two personal computers (PCs), Brainproducts Acticap system with 32 channels and an LCD monitor (BenQ XL2430-B) for stimuli presentation. Participants are seated approximately 80 cm in front of the monitor.

PC1 is used for the presentation on the LCD monitor, which is set to refresh rate of 60 Hz and its native resolution of 1920×1080 pixels. The stimuli are presented using the framework introduced in chapter 4. A stimulus can either be black or white, which can be represented by 0 or 1 in a binary sequence and is synchronized with the refresh rate of the LCD monitor. The timings of the monitor refresh cycles are synchronized with the EEG amplifier by using the parallel port.

PC2 is used for data acquisition and analysis. BCI2000 [SMH⁺04] is used as a general framework for recording the data of the EEG amplifier, and the data processing is done with MATLAB [MAT17]. The amplifier sampling rate was set to 600 Hz, resulting in 10 samples per frame/stimulus. Additionally, a TCP network connection was established to PC1 to send instructions to the presentation layer and to get the modulation patterns of the presented stimuli.

A 32 electrodes EEG layout was used, 30 electrodes were located at Fz, T7, C3, Cz, C4, T8, CP3, CPz, CP4, P5, P3, P1, Pz, P2, P4, P6, PO9, PO7, PO3, POz, PO4, PO8, PO10, O1, POO1, POO2, O2, OI1h, OI2h, and Iz. The remaining two electrodes were used for electrooculography (EOG), one between the eyes and one left of the left eye. The ground electrode (GND) was positioned at FCz and reference electrode (REF) at OZ.

8.1.5 Data acquisition

The data from the previous experiment was used, as described in section 6.1.5. Also, the same preprocessing steps were performed, as described in section 6.1.6. For the sake of completeness, the important parts will be explained again.

Initially, the participants had to perform 2 runs to generate a spatial filter (see section 6.1.6.3). The training phase was split into 3 runs, but with varying trial duration, 5 s, 4 s, and 3 s, respectively. The testing phase consists of 7 runs with a trial duration of 2 s. During the training and testing runs, the targets were modulated with fully random stimulation patterns (see section 4.3).

The spatial filter runs and test runs were used to analyze the bit-prediction accuracy relative to the number of bit-changes. In addition, to prove the findings, the participants had to perform 7 more runs, also with a trial duration of 2 s, but using the optimized set of stimulation patterns as explained.

During all runs, the inter-trial time was set to 0.75 s, and the participants had to perform 32 trials in lexicographic order using the matrix keyboard layout (see section 4.4.1).

8.1.6 Performance evaluation

As the EEG2Code model is used to analyze the optimized stimulation patterns, the same performance evaluation was used. Therefore, the accuracy of how much of the stimulation pattern can be predicted correctly, as well as the corresponding ITR was analyzed. For the latter, Eq. 2.10 is used with $N = 2$ (1 or 0), P the corresponding pattern prediction accuracy, and $T = 1/60$ as a single stimulus lasts for 1/60 s and each bit was predicted individually.

8.2 Results

For both, the used m-sequence as well as the fully random stimulation patterns, the bit-change probability was analyzed for each 15-bit window. Fig 8.1 shows the results for the m-sequence and reveals that 33 of the 63 windows have 7 or 8 bit-changes, which corresponds to a bit-change probability between 50% and 57%. The results for the random stimulation patterns are, of course, normally distributed with its maximum at 50% bit-change probability (Fig. 8.2).

It was analyzed if the bit-change probability affects the bit-prediction accuracy, by using the data from chapter 6. Fig. 8.3 shows the average bit-prediction accuracy relative to the bit-change probability. It clearly shows that the accuracy is best for bit-change probabilities between 40% and 60%, and decreases with a falling as well

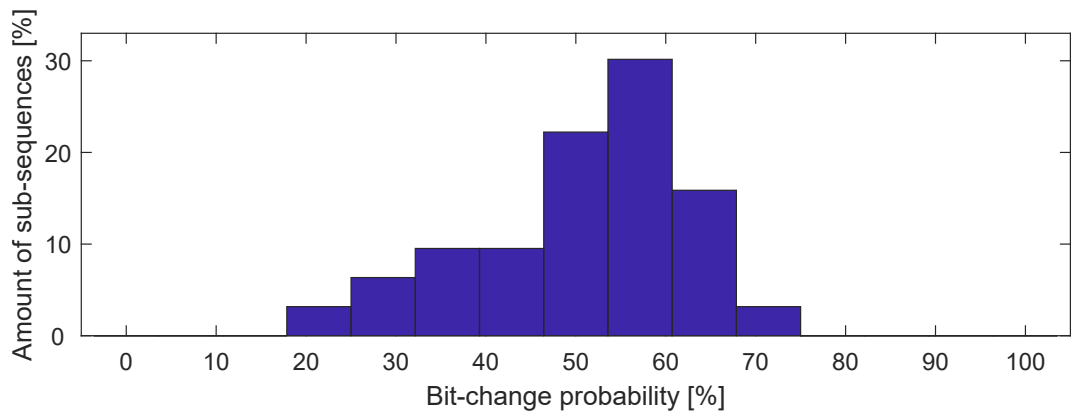


Figure 8.1: Analysis of the m-sequence. Shown are the number of 15-bit sub-sequences of the m-sequence relative to their bit-change probability.

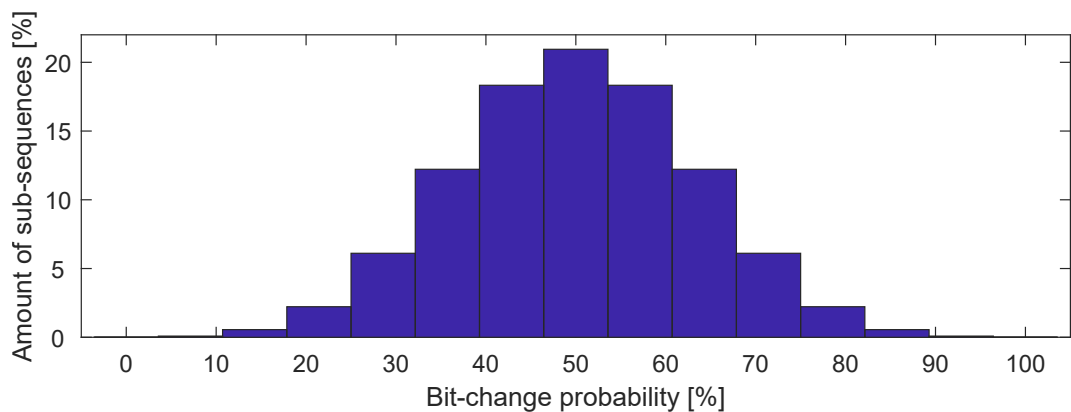


Figure 8.2: Distribution of all possible 15-bit long sequences relative to their bit-change probability.

as rising bit-change probabilities.

To further prove the effects of the bit-change probability, the participants performed trials using a set of optimized stimulation patterns which all have a bit-change probability of 50%, but not for intermediate parts (see Fig. 8.4). Table 8.1 shows the results for each participant. The fully random stimulation patterns could be predicted with an accuracy of 64.6% whereas an accuracy of 65.4% was achieved by using the optimized stimulation patterns, which corresponds to a significant ($p < 0.05$, paired t-test) increase of the ITR to 264.9 bit/min with a maximum of 481.6 bit/min.

8.3 Discussion

As explained, VEPs are evoked because of flickering stimuli, which implies that a constant "off" stimulus as well as a constant "on" stimulus does not result in a VEP,

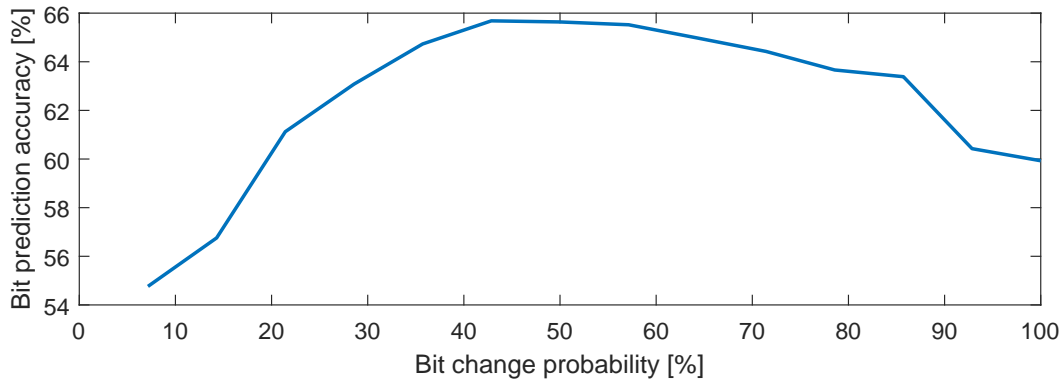


Figure 8.3: Influence of the bit-change probability using the EEG2Code model. The accuracies are averaged using all 250 ms (15 bit) sub-sequences of random modulation trials grouped by the bit-change probability. The maximum bit-prediction accuracy is reached between approximately 40% to 60%, meaning an average of 6 to 8 bit-changes.

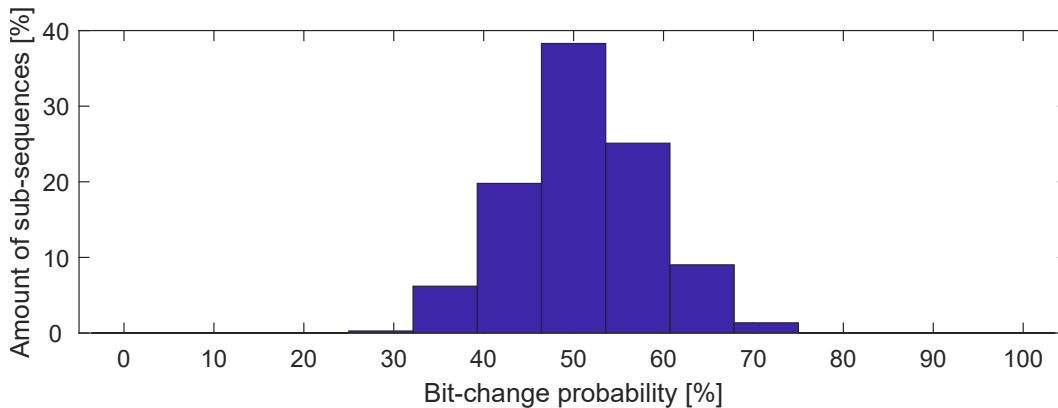


Figure 8.4: Distribution of the optimized stimulation patterns during the experiment. The optimized set consists of 15-bit patterns with a bit-change probability of 50% each. As those patterns were randomly concatenated during the experiment, a 50% bit-change probability cannot be guaranteed for the intermediate parts.

therefore, there is a lower bound. As shown by Herrmann [Her01], VEPs can be measured in the EEG only for stimulation frequencies of up to 90 Hz, which implies that there is also an upper bound. Therefore, the stimuli rate must affect the strength of VEPs.

In chapter 6 it was found that the bit-prediction accuracy using m-sequence stimulation is significantly better as the random bit-prediction accuracy, which means that the used m-sequence must have a better stimuli rate. An analysis of the m-sequences reveals that 53% of all 15-bit sequences have between 7 and 8 bit-changes, which in turn means that approximately every second bit has changed from 0 to 1 or vice versa.

An analysis of the bit-prediction accuracy using the EEG2Code model proofed, that

Table 8.1: Comparison of random and optimized stimulation.

Subject	Random stimulation		Optimized stimulation	
	ACC [%]	ITR [bpm]	ACC [%]	ITR [bpm]
S1	69.1	389.9	71.2	481.6
S2	64.5	222.4	65.1	241.5
S3	63.7	196.5	65.0	237.0
S4	65.6	257.8	66.8	299.0
S5	66.3	282.7	67.1	311.2
S6	67.1	308.2	68.5	364.9
S7	63.7	196.8	65.0	235.9
S8	60.9	124.5	60.4	113.6
S9	60.2	109.4	59.7	99.0
mean	64.6	232.0	65.4	264.9

Shown are the average results of all subjects, whereby best results are in bold font. For both, random stimulation and optimized stimulation, the accuracy (ACC) indicates how many bits of the stimulation pattern can be predicted correctly. The ITR is calculated using Eq. 2.10 with $N = 2$ and $T = 1/60s$. For the optimized stimulation, the ITRs are just for comparison.

the bit-change probability has significant effects. With bit-change probabilities between 40% and 60% the accuracy is best and decreases with a falling as well as rising bit-change probabilities. The effect was further proofed by using optimized stimulation patterns with 50% bit-change probability, which significantly increased the ITR to 264.9 bit/min. Here it is worth to mention, that the optimized stimulation patterns were randomly concatenated during the experiment, which means the 50% bit-change probability cannot be guaranteed for the intermediate parts.

Altogether, the results show that the EEG2Code model can be used to analyze the performance of different stimulation patterns or to be more precise to analyze the performance of specific properties of stimulation patterns. Another approach could be to search for stimulation patterns which can be predicted best for as many participants as possible, afterward those stimulation patterns could be used for stimulation.

Also, such an approach could be utilized for electrical stimulation for therapeutic means, where optimal stimulation parameters need to be found to evoke a certain response [WNS⁺14].

Chapter 9

Synchronous BCI control

For the models, EEG2Code and Code2EEG explained in the previous chapters, it was shown that they can decode/encode brain responses to arbitrary visual stimuli. However, this passive BCI application has no purpose for the user. In order that a user can use it for interactions, it is required to classify a specific interaction/target depending on the corresponding model prediction.

Several synchronous BCI control approaches were developed, mainly used for spelling or wheel-chair control, and have shown to be suitable for high-speed control. As already mentioned, the idea for an BCI speller was originally developed by Sutter in 1984 [Sut84], who stated that "the electrical scalp response to a modulated target is largest if the target is located within the central 1° of the visual field" and that "this makes it possible to construct a gaze-controlled keyboard". Sutter implemented a BCI speller in 1992 where he used 64 visual stimuli that were modulated by a complex stimulation pattern.

In this chapter, it will be shown that both models are suitable for active BCI control, by using them for a synchronous BCI speller. Several target classification methods will be tested, and a within-subject comparison to a state-of-the-art cVEP BCI speller was performed. Additionally, the findings of chapter 8 that optimized stimulation patterns lead to better model predictions will be analyzed in the context of active BCI control.

Parts of the results, figures and wording are already published in [NS18].

9.1 Methods

9.1.1 Target classification

For the EEG2Code model, there are two possibilities to classify the correct target. On the one hand, it is possible to use the model prediction, which is a sequence of real values, on the other hand, the predicted bit sequence can be used. Contrary, for the Code2EEG model it is only possible to use the model output (predicted brain response). An example of the EEG2Code model is depicted in Fig. 9.1.

Several distance/similarity measures were tested, which will be explained in the following.

9.1.1.1 Hamming distance

For the EEG2Code model, the first approach is to use the Hamming distance between the predicted bit sequence and the stimulation patterns of all targets. The Hamming distance $d_H(x, y)$ is the number of mismatches between two points x and y of same length n . In the case of the EEG2Code model, x is the stimulation pattern and y the predicted bit sequence, both are of equal length n .

Therefore, the Hamming distance is a measure for the bit-prediction accuracy:

$$ACC = 1 - \frac{d_H(x, y)}{n} \quad (9.1)$$

The target stimulation pattern with the smallest Hamming distance (highest bit-prediction accuracy) will be selected as the correct target.

9.1.1.2 Euclidean distance

Another distance approach is the Euclidean distance, which defines the distance in the Euclidean space. The Euclidean distance d_e between two points x and y is defined as

$$d_e(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (9.2)$$

where n is the number of dimensions. In the case of the EEG2Code model, x is the stimulation pattern and y the model prediction, which are of equal length n , depending on the trial duration. The target with the smallest distance will be selected.

The Euclidean distance could also be used for the Code2EEG model, but since the range of the model prediction is not necessarily the same as for the spatially filtered EEG, it was not used.

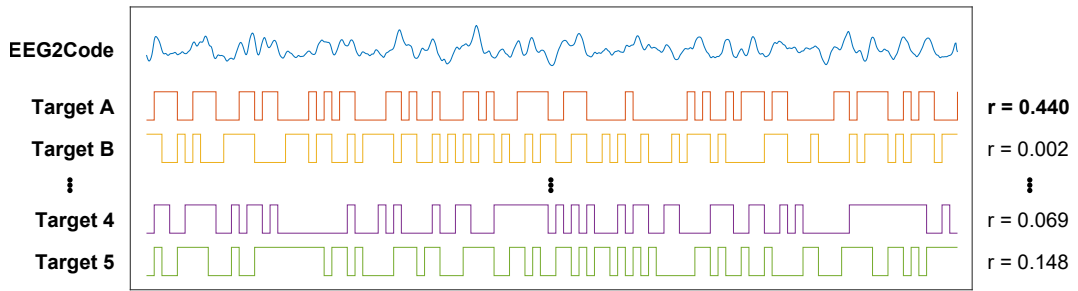


Figure 9.1: Example of the target classification using the EEG2Code model and the Pearson correlation coefficient. The blue line shows the EEG2Code model prediction and the colored lines below are the target stimulation patterns. For simplicity, only targets A, B, 4 and 5 are shown. Right to each stimulation pattern is the corresponding correlation coefficient. The correlation is highest for target A, therefore, it is chosen as the correct target.

9.1.1.3 Correlation

As explained in section 2.6.3, the correlation is a similarity measure, therefore, it is also suitable for target classification. The method is used for both models. For the EEG2Code model, the correlation coefficient is calculated between the model prediction and the target stimulation pattern. For the Code2EEG model, it is calculated between the predicted brain response and the spatially filtered EEG.

9.1.2 Hardware & Software

The same setup was used as explained in section 6.1.4. For the sake of completeness, it will be explained again.

The system consists of a g.USBamp (g.tec, Austria) EEG amplifier, two personal computers (PCs), Brainproducts Acticap system with 32 channels and an LCD monitor (BenQ XL2430-B) for stimuli presentation. Participants are seated approximately 80 cm in front of the monitor.

PC1 is used for the presentation on the LCD monitor, which is set to refresh rate of 60 Hz and its native resolution of 1920×1080 pixels. The stimuli are presented using the framework introduced in chapter 4. A stimulus can either be black or white, which can be represented by 0 or 1 in a binary sequence and is synchronized with the refresh rate of the LCD monitor. The timings of the monitor refresh cycles are synchronized with the EEG amplifier by using the parallel port.

PC2 is used for data acquisition and analysis. BCI2000 [SMH⁺04] is used as a general framework for recording the data of the EEG amplifier, and the data processing is done with MATLAB [MAT17]. The amplifier sampling rate was set to 600 Hz, resulting in 10 samples per frame/stimulus. Additionally, a TCP network connection

was established to PC1 to send instructions to the presentation layer and to get the modulation patterns of the presented stimuli.

A 32 electrodes EEG layout was used, 30 electrodes were located at Fz, T7, C3, Cz, C4, T8, CP3, CPz, CP4, P5, P3, P1, Pz, P2, P4, P6, PO9, PO7, PO3, POz, PO4, PO8, PO10, O1, POO1, POO2, O2, OI1h, OI2h, and Iz. The remaining two electrodes were used for electrooculography (EOG), one between the eyes and one left of the left eye. The ground electrode (GND) was positioned at FCz and reference electrode (REF) at OZ.

9.1.3 Data acquisition

The data acquisition explained in chapters 6-8, was acquired through the synchronous BCI experiment explained in this chapter, whereby for the online results the EEG2Code model and the correlation coefficients were used to classify the targets. For the sake of completeness, the important parts will be explained again.

Initially, the participants had to perform 2 runs to generate a spatial filter (see section 6.1.6.3). The training phase was split into 3 runs, but with varying trial duration, 5 s, 4 s, and 3 s, respectively. The testing phase was split into 14 runs with a trial duration of 2 s. The runs were alternated using fully random stimulation patterns (see section 6.1.3) and optimized stimulation patterns (see section 8.1.3).

During all runs, the inter-trial time was set to 0.75 s, and the participants had to perform 32 trials in lexicographic order using the matrix keyboard layout (see section 4.4.1).

9.1.4 Comparison to a cVEP BCI speller

To compare the EEG2Code BCI speller to a state-of-the-art cVEP BCI speller method [SRB12a], a within-subject comparison was performed using the m-sequence trials of the spatial filter runs. The cVEP analysis was performed as described in [SRB12a] using a one-class support-vector-machine (OCSVM), for this the 192 m-sequence trials (2 runs \times 32 trials \times 3 m-sequence cycles) were analyzed using a randomized 10-fold cross-validation (CV). For the EEG2Code model trained on the random stimulation patterns, all 192 trials were classified. Furthermore, a randomized 10-fold cross-validation (CV) analysis was performed by using the m-sequence trials also to train the EEG2Code model. To clarify, this means that the first EEG2Code model was not trained on m-sequences, whereas the second EEG2Code model is only trained on m-sequences.

9.1.5 Performance evaluation

To analyze the performance of the synchronous BCI control, the accuracy of the target prediction is calculated using Eq. 2.9. In addition to the accuracies, the corresponding ITRs were calculated using Eq. 2.10 with $N = 32$, P the corresponding target prediction accuracy, and $T = 2.75$ s as a trial lasts for 2 s and the inter-trial time was 0.75 s.

9.2 Results

9.2.1 Comparison of the different target classification methods

For the EEG2Code model, it is possible to use different methods to classify the correct target based on the model prediction. All 14 test runs (448 trials in total) were used to analyze the different performances. Table 9.1 shows detailed results for each participant.

During the online BCI control, the correlation coefficient was used, which resulted in an average target prediction accuracy of 97.9%. In a further offline analysis, the synchronous BCI control was simulated using the Euclidean-distance for target classification, which resulted in a significantly ($p < 0.05$, paired t-test) reduced accuracy of 95.4%. The third used method was the Hamming distance between the predicted stimulation pattern and the target stimulation patterns which also resulted in a sig-

Table 9.1: EEG2Code BCI control using different target classification methods.

Subject	Correlation ACC [%]	Euclidean ACC [%]	Hamming ACC [%]
S1	100.0	100.0	99.3
S2	99.4	98.9	97.3
S3	97.8	96.0	94.0
S4	100.0	99.6	98.0
S5	99.6	99.3	97.1
S6	100.0	99.8	98.4
S7	98.9	97.8	93.5
S8	98.0	87.3	71.7
S9	87.7	79.7	69.2
mean	97.9	95.4	90.9

Target prediction accuracies (ACC) using the EEG2Code model with different target classification methods: correlation coefficient, Euclidean distance, and Hamming distance. All runs with random stimulation patterns as well as optimized stimulation patterns were analyzed (448 trials in total).

nificantly ($p < 0.05$, paired t-test) reduced accuracy of 90.9%.

9.2.2 Comparison of EEG2Code and Code2EEG

In addition to the EEG2Code model, the Code2EEG was tested in an offline analysis. For the target classification, the correlation coefficient was calculated between the spatially filtered EEG and the predicted EEG of each target stimulation pattern, whereby all 14 test runs were analyzed (448 trials in total). As shown above, the EEG2Code model achieved an average target prediction accuracy of 97.9% (ITR: 104.4 bit/min) using the correlation coefficient. The Code2EEG model performs significantly ($p < 0.05$, paired t-test) worse with an average accuracy of 93.9% (ITR: 96.8 bit/min). The ITRs were calculated including an inter-trial time of 0.75 s. Detailed results for each participant are shown in Table 9.2.

Table 9.2: BCI control comparison of EEG2Code and Code2EEG.

Subject	EEG2Code		Code2EEG	
	ACC [%]	ITR [bpm]	ACC [%]	ITR [bpm]
S1	100.0	109.1	100.0	109.1
S2	99.4	107.2	96.2	99.9
S3	97.8	103.3	93.5	94.5
S4	100.0	109.1	99.1	106.5
S5	99.6	107.7	98.9	106.0
S6	100.0	109.1	99.3	107.1
S7	98.9	106.0	98.0	103.8
S8	98.0	103.8	80.6	72.6
S9	87.7	84.1	79.9	71.6
mean	97.9	104.4	93.9	96.8

Accuracies (ACC) and information transfer rates (ITR) using the EEG2Code and Code2EEG model. For both models the runs with random stimulation patterns as well as optimized stimulation patterns were analyzed (448 trials in total). ITRs are calculated using Eq. 2.10 with $N = 32$ and $T = 2.75s$ (including an inter-trial time of 0.75s).

9.2.3 Comparison of random and optimized stimulation patterns

Since the EEG2Code model outperformed the Code2EEG model, it is used to further analyze the target classification performance difference between fully random stimulation patterns and the optimized stimulation patterns. The findings of chapter 8 suggest an improved performance.

During each run the participants had to spell each letter in lexicographic order, meaning 32 trials per run for a total of 224 trials for each stimulation type. Table 9.3 shows the target prediction accuracies and the corresponding ITRs, including the inter-trial time of 0.75 s. When using fully random stimulation patterns, the average accuracy of target selection is 97.8%, which corresponds to 103.9 bpm with the used trial duration of 2 s. As shown in chapter 8, using optimized modulation patterns resulted in a better bit-prediction accuracy, this can also be confirmed for the target classification, which results in an accuracy of 98.0% (104.9 bpm). But due to a ceiling effect, the difference between random and optimized stimulation patterns is not significant. Interestingly, only participant S9 achieved a better performance using fully random stimulation patterns (84.8% vs. 90.6%).

Table 9.3: BCI control comparison of random and optimized stimulation.

Subject	optimized		random	
	ACC [%]	ITR [bpm]	ACC [%]	ITR [bpm]
S1	100.0	109.1	100.0	109.1
S2	99.6	107.7	99.1	106.5
S3	99.1	106.5	96.4	100.4
S4	100.0	109.1	100.0	109.1
S5	100.0	109.1	99.1	106.5
S6	100.0	109.1	100.0	109.1
S7	99.6	107.7	98.2	104.3
S8	99.1	106.5	96.9	101.3
S9	84.8	79.3	90.6	89.2
mean	98.0	104.9	97.8	103.9

Accuracies (ACC) and information transfer rates (ITR) of the online BCI experiment (EEG2Code model). Shown are the average results of all subjects using random stimulation patterns and optimized stimulation patterns. ITRs are calculated using Eq. 2.10 with $N = 32$ and $T = 2.75s$ (including an inter-trial time of 0.75s).

9.2.4 Comparison to a cVEP BCI speller

A within-subject comparison to a state-of-the-art cVEP BCI speller method [SRB12a] was performed using the m-sequence trials of the spatial filter runs. The results are listed in Table 9.4. By using the EEG2Code model trained on trials using fully random stimulation patterns, an average target prediction accuracy of 89.7 % was reached, which corresponds to an ITR of 155.3 bpm (with $N = 32$ and $T = 1.55s$). It should be noted that the model was not trained on any of the m-sequence trials. The EEG2Code model trained on only m-sequence trials performs better with an accuracy of 90.8 % (ITR: 158.6 bpm). Using the OCSVM method the average accuracy is 93.2 % (ITR: 166.5 bpm), whereas the

results are not significantly better as the results of the EEG2Code model trained on only m-sequences ($p > 0.05$, paired t-test).

Table 9.4: Performance comparison using m-sequence stimulation for cVEP BCI.

Subject	EEG2Code		OCSVM [SRB12a]
	trained on random sequences	trained on m-sequences	trained on m-sequences
S1	100.0%	100.0%	100.0%
S2	94.8%	97.9%	98.9%
S3	50.0%	39.0%	53.7%
S4	100.0%	99.5%	100.0%
S5	98.4%	98.4%	98.4%
S6	95.3%	96.3%	96.8%
S7	97.4%	98.4%	97.9%
S8	86.5%	91.6%	93.2%
S9	84.9%	95.8%	99.5%
mean	89.7%	90.8%	93.2%

Shown are the accuracies (ACC) of the target prediction on data where m-sequences were used for stimulation. The EEG2Code model was trained once on random stimulation patterns and once on the m-sequence trials using a randomized 10-fold cross-validation (CV). The results are compared to the results using the one-class support vector machine (OCSVM) also by using a randomized 10-fold CV. The results are averaged over 192 trials with a duration of 1.05 s.

9.3 Discussion

In this chapter, it was shown that the EEG2Code, as well as the Code2EEG model, can be used for BCI control. When using the EEG2Code model with correlation coefficients for target selection, as done during the online BCI control, an average accuracy of 97.9% was achieved, which corresponds to an ITR rate of 108 bit/min with the used trial time of 2 s and the inter-trial time of 0.75 s. Using the Euclidean distance measure resulted in 95.4% which is significantly less ($p < 0.05$, paired t-test) compared to using the correlation coefficients. Using the Hamming distance, resulted in a significantly ($p < 0.05$, paired t-test) worst performance (90.9%), which could be explained by the loss of information due to the rounding of the model prediction. Altogether, the results suggest using the correlation coefficient for target selection to achieve the best classification accuracies.

A direct comparison of the EEG2Code and Code2EEG model reveals a significantly ($p < 0.05$, paired t-test) better performance using the EEG2Code model (97.9% vs. 93.9%). This can be explained by the fact, as discussed in chapter 7, that the

Code2EEG model only predicts the evoked response in the EEG, and not the noise present in the EEG, which in turn reduces the correlation coefficient by comparing it to the spatially filtered EEG on a single trial basis. Contrary, the EEG2Code model prediction is compared to the (noise-free) stimulation patterns. But although the performance is reduced, with an ITR of 96.8 bit/min, it is more than twice as fast as the only comparable work by Thielen *et al.* [TvdBFD15], which achieved an ITR of 38.1 bit/min at an even lower accuracy of 86%.

As shown in chapter 8, the EEG2Code model achieves a significantly better performance by using the optimized stimulation patterns instead of fully random stimulation patterns. The results can be confirmed for BCI control, the optimized stimulation patterns resulted in a better target prediction accuracy (98% vs. 97.8%), but because of the ceiling effect, the results are not significant. Furthermore, only participant S9 achieved a better performance using the fully random stimulation patterns, which, however, also shows up in the results of the bit-prediction accuracy. Interestingly, S8 also achieved a better bit-prediction accuracy using the random stimulation, but a better target prediction accuracy using the optimized stimulation patterns.

A further conceivable optimization for the BCI control could be the adaption of the stimulation pattern. Starting with fully random patterns, the bit-prediction accuracy of a defined length will be analyzed continuously and will be rated, for example, by the accuracy itself. In the following the stimulation will be adapted, the higher the bit-prediction accuracy of a pattern the more likely it will be presented. This could lead to more optimized patterns. Additionally, also an adaptive train would be possible, for example, a trial could be added to the training set if the stimulation pattern can be predicted with a certain bit-prediction accuracy.

A within-subject comparison of the EEG2Code model to a state-of-the-art cVEP BCI speller reveals a comparable performance. Although the EEG2Code model has slightly reduced performance (90.8% vs. 93.2%), the difference is not significant ($p > 0.05$, paired t-test).

The presented online BCI performance is below the best results reported for a cVEP BCI (144 bpm [SRB12b]) or an SSVEP BCI (267 bpm [CWN⁺15]). However, the BCI used in the online setup was not optimized to achieve a high ITR, but to be comfortably usable by a BCI-naive person and to get the required data for the offline analyses. The optimization of the parameters will be addressed in chapter 8.

Chapter 10

Increasing the number of targets and varying trial durations

As mentioned in chapter 6, recent VEP BCIs are either based on the original idea of Sutter [Sut84] and use complex stimulation patterns to elicit a cVEP [SRB12b, BGW⁺11], or they are based on visual stimuli modulated by specific frequencies which evoke SSVEPs [CWN⁺15]. Although both methods differ in how the stimulation pattern is constructed, both methods depend on the construction of a stimulus-specific template, restricting the number of possible targets. For example, a state-of-the-art cVEP BCI speller uses a 63-bit m-sequence, which is circularly shifted by 2 bits for each target, thereby allowing a total of 32 targets. To increase the number of targets, the shift must be decreased, or the sequence length must be increased, either decreasing performance or speed. Another approach is to group several targets and use multiple m-sequences for each group [WLG⁺18, LWL18], but this approach is also limited as the number of equal-sized m-sequences is limited. In case of SSVEP BCIs, there are also some approaches to increase the number of targets, for example by using multiple frequencies sequential coding [ZXL⁺12] or by using multi-phase cycle coding [TZ15], but both show a reduced information transfer rates.

In this chapter, the two most significant advantages of the EEG2Code and the Code2EEG model are addressed, the nearly unlimited number of targets as well as the variable trial duration. In an offline analysis, the number of targets was increased up to half a million targets, and the trial durations were varied between 0.5 s and 2 s.

Parts of the results, figures and wording are already published in [NS18].

10.1 Methods

10.1.1 Varying trial duration and number of targets

The EEG2Code (chapter 6) and Code2EEG (chapter 7) model are based on a sliding window approach, which means the prediction does not require the complete data but a 250 ms window, therefore, the trial duration can be varied. In an offline analysis, the BCI performance was tested with varying the trials durations from 0.5 s to 2 s.

Furthermore, the models are trained on fully random stimulation patterns, therefore, it is not required to use the same stimulation patterns for training and testing, but arbitrary stimulation patterns can be used which in turn allows varying the number of targets. Additional targets simply get another random stimulation pattern.

With a trial length of 2 s and 60 Hz refresh rate, there are $2^{120} = 1.33 \cdot 10^{36}$ different stimulation patterns when using fully random stimulation patterns and $150^8 = 2.56 \cdot 10^{17}$ when using the optimized stimulation patterns, which therefore are the upper bounds for the number of targets with 2 s trial duration. As mentioned in chapter 9, the layout had 32 targets, which means the model prediction had to be compared to the 32 stimulation patterns to classify the correct target. In an offline analysis, the synchronous BCI control was simulated with up to 500,000 targets in steps of 50 targets, which means the model prediction has also to be compared to the additional simulated targets, which essentially makes this a multiclass classification problem with up to 500,000 classes.

10.1.2 Hardware & Software

The same setup was used as explained in section 9.1.2. For the sake of completeness, it will be explained again.

The system consists of a g.USBamp (g.tec, Austria) EEG amplifier, two personal computers (PCs), Brainproducts Acticap system with 32 channels and an LCD monitor (BenQ XL2430-B) for stimuli presentation. Participants are seated approximately 80 cm in front of the monitor.

PC1 is used for the presentation on the LCD monitor, which is set to refresh rate of 60 Hz and its native resolution of 1920×1080 pixels. The stimuli are presented using the framework introduced in chapter 4. A stimulus can either be black or white, which can be represented by 0 or 1 in a binary sequence and is synchronized with the refresh rate of the LCD monitor. The timings of the monitor refresh cycles are synchronized with the EEG amplifier by using the parallel port.

PC2 is used for data acquisition and analysis. BCI2000 [SMH⁺04] is used as a general framework for recording the data of the EEG amplifier, and the data processing is done with MATLAB [MAT17]. The amplifier sampling rate was set to 600 Hz,

resulting in 10 samples per frame/stimulus. Additionally, a TCP network connection was established to PC1 to send instructions to the presentation layer and to get the modulation patterns of the presented stimuli.

A 32 electrodes EEG layout was used, 30 electrodes were located at Fz, T7, C3, Cz, C4, T8, CP3, CPz, CP4, P5, P3, P1, Pz, P2, P4, P6, PO9, PO7, PO3, POz, PO4, PO8, PO10, O1, POO1, POO2, O2, OI1h, OI2h, and Iz. The remaining two electrodes were used for electrooculography (EOG), one between the eyes and one left of the left eye. The ground electrode (GND) was positioned at FCz and reference electrode (REF) at OZ.

10.1.3 Data acquisition

The same data is used as explained in section 9.1.3, but only the test runs using the optimized stimulation patterns were used, as the model prediction performed better compared to fully random stimulation patterns (see chapter 8). For the sake of completeness, the important parts will be explained again.

Initially, the participants had to perform 2 runs to generate a spatial filter (see section 6.1.6.3). The training phase was split into 3 runs, but with varying trial duration, 5 s, 4 s, and 3 s, respectively. The testing phase consists of 7 runs with a trial duration of 2 s.

During all runs, the inter-trial time was set to 0.75 s, and the participants had to perform 32 trials in lexicographic order using the matrix keyboard layout (see section 4.4.1). During the training runs the targets were modulated with fully random stimulation patterns (see section 6.1.3) and during the test runs with the optimized stimulation patterns (see section 8.1.3).

10.1.4 Performance evaluation

Since the analysis simulates a synchronous BCI control, the accuracy of the target prediction (Eq. 2.9), as well as the corresponding ITRs, were calculated. For the latter Eq. 2.10 is used with N the corresponding number of simulated targets, P the corresponding target prediction accuracy, and T the used trial duration including an inter-trial time of 0.5 s.

10.2 Results

The results are based on the 7 runs with optimized stimulation patterns, as it was shown in the previous chapters, that they perform better compared to fully random stimulation. Furthermore, for ITR calculation an inter-trial-time of 0.5 s was used, as this is sufficient for experienced participants.

Using the EEG2Code model with 32 targets, varying the length of the trials from 0.5 s to 2 s shows that average accuracy increases with longer trials, but the ITR reaches its optimum of 154.3 bpm with a trial duration of 1 s. It is worth mentioning that S1 achieved an ITR of 231.1 bpm using 600 ms trial duration which corresponds to an accuracy of 92.41 %.

The Code2EEG model is generally less accurate, reaching only 94.4 % with 2 s trials, but still has an optimum average ITR of 146.6 bpm with a trial length of 0.75 s. More detailed results for the EEG2Code model are shown in Fig. 10.1 and for the Code2EEG model in Fig.10.2.

As both models are not limited to fixed stimulation patterns but work with arbitrary patterns, the BCI was simulated with up to 500,000 targets in steps of 50 targets. Fig. 10.3 shows the accuracies and ITRs of the EEG2Code model relative to the number of targets. The averaged optimum ITR was at 235.3 bpm using 71,930 targets, although the optimum varied largely between subjects.

For subject S1 the ITR was still increasing at half a million targets with an accuracy of 96.3% and an ITR of 432 bpm. Furthermore, S1 achieved an accuracy of 100% for up to 29,500 targets. In Fig. 10.5 are detailed results for S1 showing the target variation for all trial durations, revealing a maximum ITR of 474.5 bpm using 472,700 targets and a trial duration of 1.5 s, which would be, to the best of our knowledge, the highest reported ITR of an offline BCI.

Using the Code2EEG model, the maximum average ITR of 183.8 bpm is reached with 9,600 targets with an average accuracy of 64.93%. Adding more targets decreases the ITR for most of the participants. Detail results for the Code2EEG model can be found in Fig. 10.4.

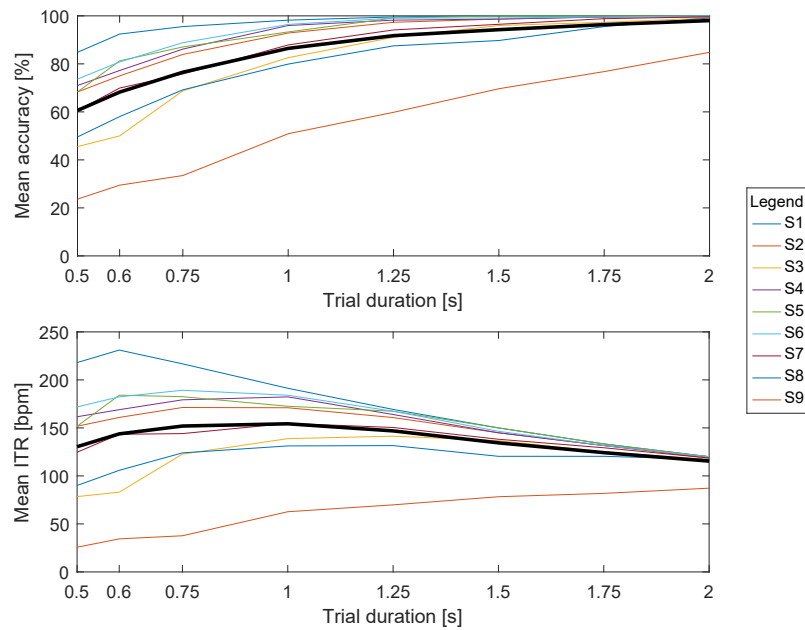


Figure 10.1: BCI performance using the EEG2Code model under different conditions. Shown are the accuracies and ITRs (Eq. 2.10 with $N = 32$ the number of targets and T the trial duration including 0.5 s inter-trial time). Each colored line is one subject and the thick black line represents the mean of all subjects.

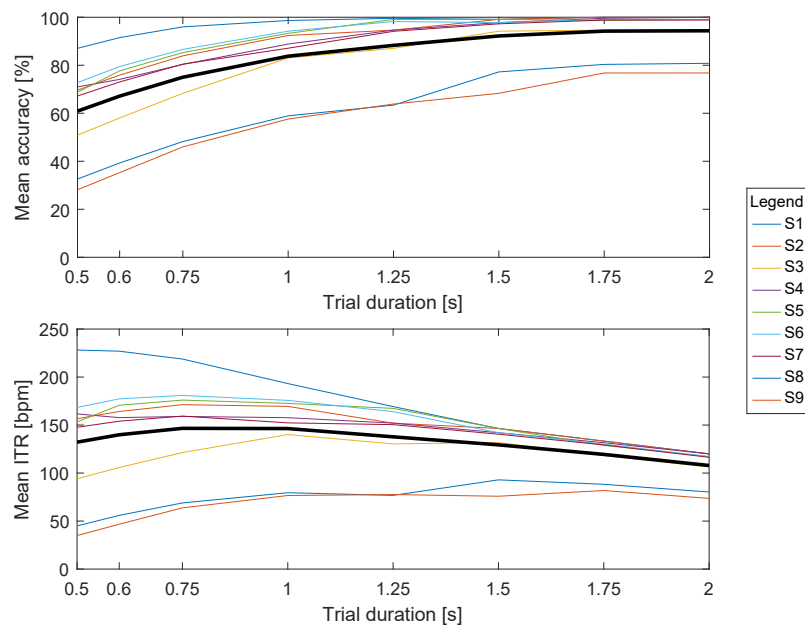


Figure 10.2: BCI performance using the Code2EEG model under different trial durations. Shown are the accuracies and ITRs (Eq. 2.10 with $N = 32$ the number of targets and T the trial duration including 0.5 s inter-trial time). Each colored line is one subject and the thick black line represents the mean of all subjects.

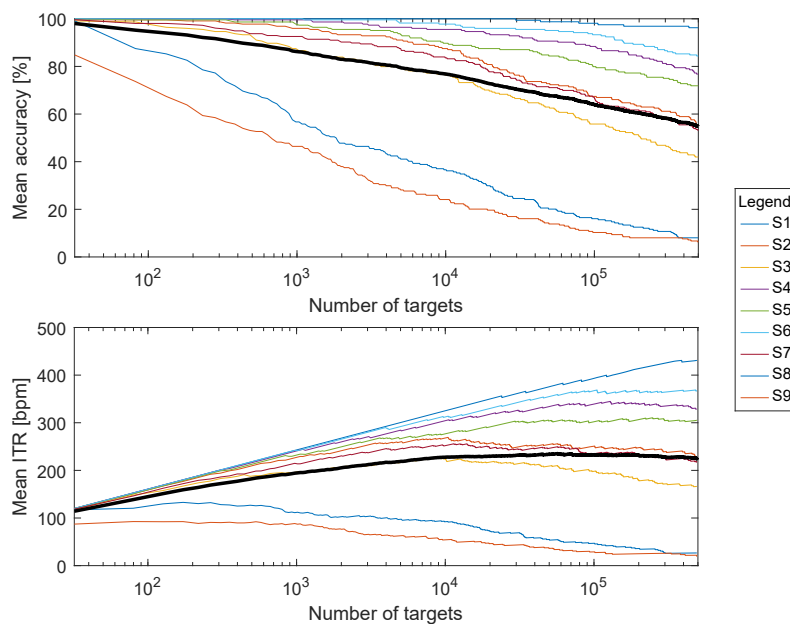


Figure 10.3: BCI performance using the EEG2Code model with a different number of targets. Shown are the accuracies and ITRs (Eq. 2.10 with N the number of targets and $T = 2.5$ the trial duration including 0.5 s inter-trial time). Each colored line is one subject and the thick black line represents the mean of all subjects.

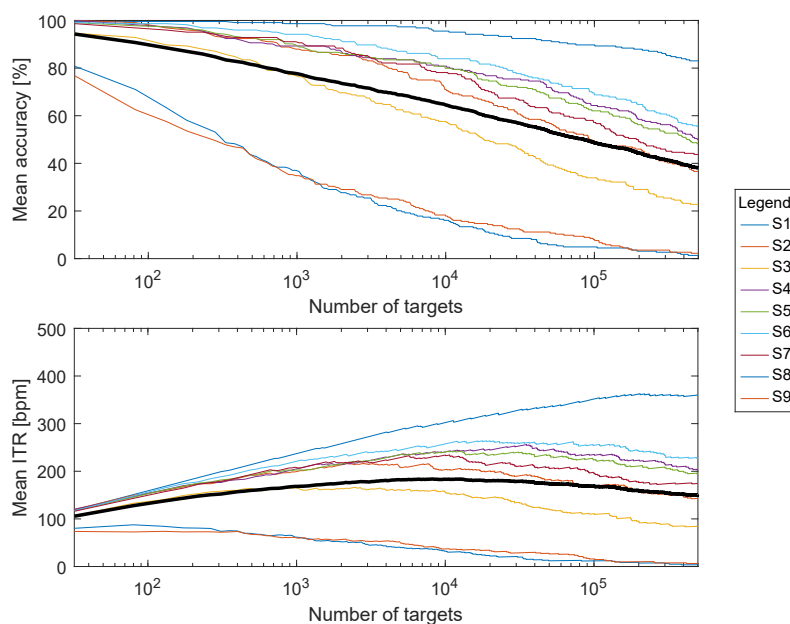


Figure 10.4: BCI performance using the Code2EEG model with a different number of targets. Shown are the accuracies and ITRs (Eq. 2.10 with N the number of targets and $T = 2.5$ the trial duration including 0.5 s inter-trial time). Each colored line is one subject and the thick black line represents the mean of all subjects.

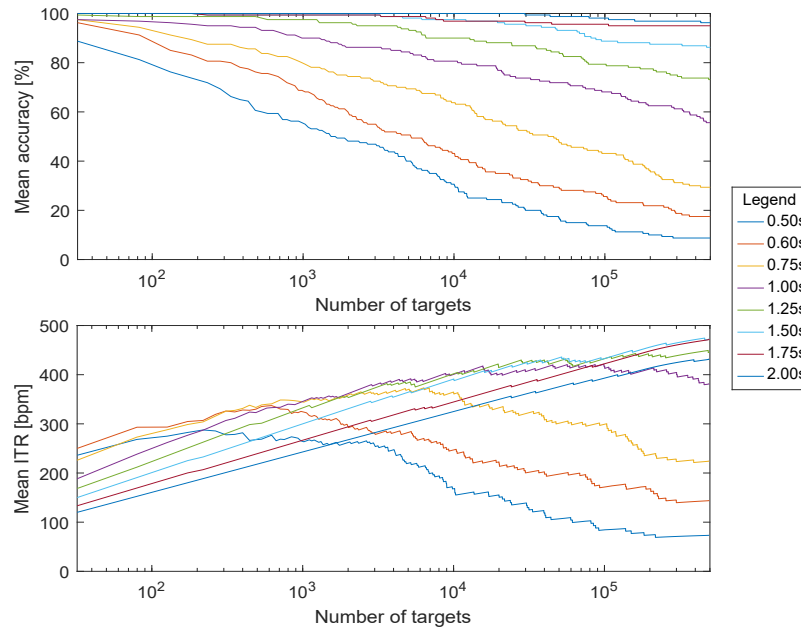


Figure 10.5: BCI performance of the best subject (S1) using the EEG2Code model. Shown are the accuracies and ITRs (including 0.5 s inter-trial time) using varying number of targets for varying trial durations. Each colored line represents a different trial duration. The maximum ITR of 474.5 bpm is reached using a trial duration of 1.5 s and 472,700 additional targets.

10.3 Discussion

As discussed, the BCI performance shown in chapter 9 was not optimized to achieve a high ITR, but to be comfortably usable by a BCI-naive person and to get the required data for the offline analysis. The current results show that the parameters can be optimized to achieve higher ITRs.

By reducing the trial duration to 1 s, the EEG2Code model achieved an average ITR of 154.3 bpm and up to 231.1 bpm, which is a comparable performance like a state-of-the-art cVEP BCI speller (144 bit/min) [SRB12b]. For shorter as well as longer trial durations, the ITRs are lower.

Beside different trial durations, the BCI performance was analyzed by simulating an increasing number of targets. Using the EEG2Code model with 1000 targets, the average accuracy is still around 90 % and goes down to around 55 % for half a million targets, with the best subject still achieving >95 % accuracy. As the *Dictionary of Chinese Variant Form* compiled by the Taiwan (ROC) Ministry of Education in 2004 contains 106,230 individual characters, this BCI approach would theoretically allow selecting each character of that alphabet individually. Although this thought is purely theoretical as there are practical limitations, like both displaying or finding a target out of such a high number of targets. Furthermore, increasing the number

of targets and not increasing the monitor results in smaller targets which will also reduce the performance. Nevertheless, those theoretical ITRs are still interesting as they provide a lower bound for the maximum amount of information that is contained in EEG data.

For the best subject S1, a detailed analysis was performed by using the increased number of targets for each tested trial duration. A theoretical maximum ITR of 474.5 bpm was found by using a trial duration of 1.5 s. It is likely that the maximum is even higher because only results for up to 500,000 targets were simulated and the ITR was still rising at that point.

Compared to other types of BCIs, such a high number of possible targets is a unique feature of the EEG2Code BCI. There are other approaches increasing the number of targets, like using multiple m-sequences in cVEP BCIs [WLG⁺18, LWL18] or by using a sequential coding strategy in SSVEP BCIs [ZXL⁺12, TZ15]. But both are still limited, and there is, to the best of my knowledge, no method allowing a virtually unlimited number of possible targets (e.g., 96.3 % accuracy for 500,000 targets with a trial length of 2 s for the best subject S1).

Although it was shown in chapter 9 that the EEG2Code model outperforms the Code2EEG model, the analysis was done for both models, to confirm the findings. It can be confirmed that the Code2EEG model performs significantly worse for all trial durations as well as for the varying number of targets. As discussed in chapter 7 participant S8 achieved a similar correlation like S3 (0.344 vs. 0.364, Table 7.1) using the Code2EEG model, but the bit-prediction accuracy of S3 is clearly increased (63.7 bit/min vs. 60.9 bit/min). This is also the case for the BCI control, S8 performs significantly worse by using the Code2EEG model. Consequently, the EEG2Code model is superior to the Code2EEG model, at least for BCI control.

Altogether, the EEG2Code model offers full flexibility for BCI application design as the number of targets as well as the trial durations can be varied freely. Especially the latter allows using the method for an asynchronous BCI, which will be addressed in the following chapter.

Chapter 11

Asynchronous BCI control

Although recent BCI spellers [SRB12b,CWN⁺15] show high communication speed, they are based on synchronous control, which means that commands are executed in a certain time interval controlled by the BCIs. However, those BCIs are not suitable for real-world applications as they cannot differentiate between control and non-control state and will give a random output if a user is taking a break to think or does not want to control the BCI for other reasons. Therefore, a practical BCI should be asynchronous, or also called self-paced, and should be able to identify the user's intent to control the system. The BCI has to distinguish efficiently between the intentional control (IC) state and the non-control (NC) state, which has been tackled by several methods [ASA⁺11,PPS11,PFH⁺15,MNK⁺17,PMTA09,XLX⁺13,ZGW08,LPWY13,ABG18,ST18,Cec10,ZTLZ17,DMPL11,YLP⁺14,PAB⁺10].

Unfortunately, the comparison of those methods is difficult, partly because the term "asynchronous" is not uniformly defined, sometimes it is used for early stopping methods disregarding the NC state evaluation and sometimes for distinguishing between IC and NC. It is also worth to note that no state-of-the-art criteria exist what a user should/could do during the NC state, which makes the comparability even more difficult. Furthermore, there are trial-based approaches with fixed trial durations classifying each trial as IC or NC [ASA⁺11,PPS11,PFH⁺15,MNK⁺17,PMTA09,XLX⁺13], and on the other hand, there are continuous classifications without fixed trial durations [ZGW08,LPWY13,ABG18]. However, compared to synchronous methods, the current asynchronous BCIs are substantially slower, and all of them did not achieve a reliable NC state recognition, meaning the BCI still executes random commands during the NC state, which decreases the user experience.

In this chapter, the EEG2Code model is used to implement an asynchronous BCI speller. It achieved a robust NC state detection under different conditions, like reading or looking at another monitor, and still allows high-speed BCI control.

Parts of the results, figures and wording are already published in [NS19a].

11.1 Methods

11.1.1 Asynchronous BCI control

Like for the synchronous BCI control, a method is required to choose the correct target out of others based on the EEG2Code model prediction y , and the classification should only be done if y arises from one of the stimulation patterns. As shown in chapter 6 y highly correlates with its corresponding stimulation pattern, and it was shown in chapter 9 that the correlation works best for target classification.

The current approach for asynchronous control is designed to have variable trial durations, but as the correlation coefficient does not take the length of y into account, the corresponding p-values (under the hypothesis that the correlation coefficient is greater than zero) are used instead for target classification. As mentioned, a classification should only be done if y arises from one of the stimulation patterns. Since the EEG2Code model prediction reaches just a certain precision, a suitable p-value threshold is required which will classify a target with a certain probability.

For better understanding, the complete procedure is depicted in Fig. 11.1. As mentioned, the approach is not based on a fixed trial length, therefore, the p-values are calculated continuously using sub-trial windows. Those windows are called sub-trials, on the hand to avoid mix-ups with the windows used for the EEG2Code model prediction, and on the other hand, only a part of the current trial data is used once a user-specific maximum trial length is reached. If a sub-trial window reaches its maximum length, the window will be shifted, which means the beginning of the trial will be discarded. As soon as a p-value is lower than a user-specific threshold (calculated beforehand), the trial stops and the corresponding target t will be selected. After an inter-trial time of 0.5 s the next trial starts. If none of the p-values reach the threshold, the trial will continue, which is the case during a non-control phase or if a target cannot be classified with a certain probability.

11.1.2 Threshold determination

As different participants do not have the same VEP responses, a user-specific threshold is determined. Furthermore, for better-performing participants, shorter sub-trial durations are sufficient, which is why also a user-specific upper sub-trial duration was determined. It should be noted that longer sub-trial durations should generally lead to a better target prediction, but if the sub-trial duration is too long, the classification after a non-control phase will take longer.

It was assumed that a minimum sub-trial length of 500 ms and a maximum sub-trial length of 3000 ms should be sufficient for all participants. Based on this assumption, each trial of the training runs was split into 50 randomly chosen sub-trials for each length between 500 ms and 3000 ms (in steps of 250 ms). This results in a bucket of

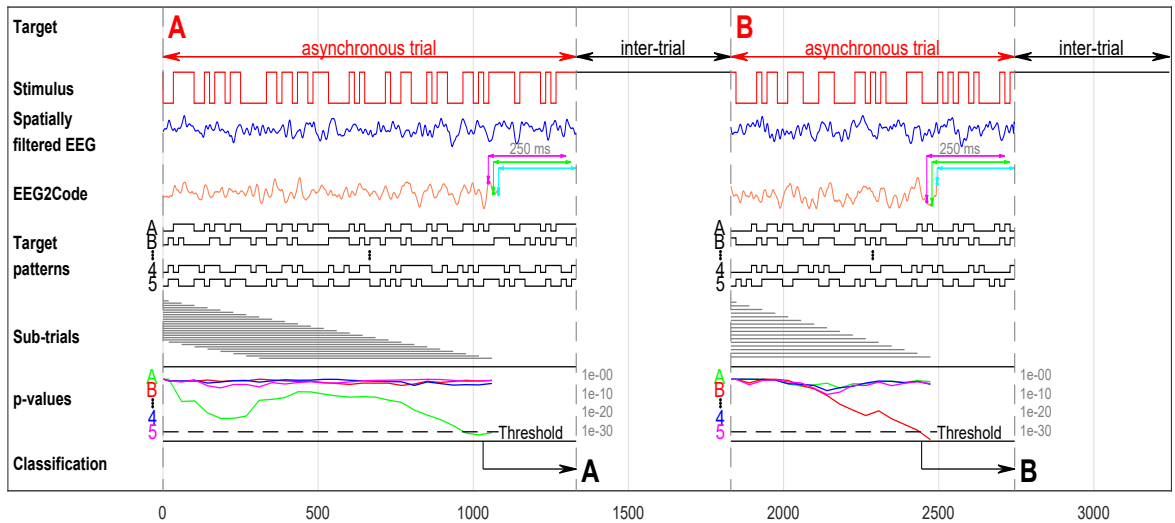


Figure 11.1: Asynchronous target classification. During a trial, the participant has to focus a target. A spatial filter is applied to the measured EEG. For each 250 ms window (slided sample-wise) of the spatially filtered EEG signal (blue line), the EEG2Code model predicts a real value (orange line) which highly correlates to the stimulation pattern. For simplicity, it is only shown for 3 exemplary windows (magenta, green, cyan). Note that the model prediction is delayed by 250 ms because of the sliding window approach. The resulting model prediction is now continuously compared to the stimulation patterns of all targets, not by using the whole trial, but sub-trials (grey lines). It was specified that a sub-trial has a user-specific maximum length (calculated beforehand), once the maximum length is reached, the sub-trial window will be shifted, meaning the beginning of the trial will be discarded. The comparison is done by calculating the p-values with the hypothesis that the correlation coefficient is greater than zero, for simplicity it is only shown for targets A, B, 4 and 5. If at any time one of the p-values is lower than a user-specific threshold (calculated beforehand), the trial stops and the corresponding target will be selected. After an inter-trial time of 0.5 s the next trial starts.

50 sub-trials (per trial) for each of those sub-trial durations. For each of those sub-trials, the corresponding p-values were calculated as explained above. Furthermore, it was determined if the trial can be predicted correctly using only the sub-trial data.

The p-value threshold was defined as the p-value of the first percentile out of all sub-trials that lead to a misclassification. For better understanding, this means that 99% of all misclassifications should be excluded by such a threshold. The maximum sub-trial duration is defined as the duration where 99% of all correctly classified sub-trials using that duration have lower p-values as the threshold. If this is not the case for any sub-trial duration, it is set to 3000 ms.

To optimize the threshold for NC state detection, the participants had to perform a single 2-minute trial where they had to look below the monitor. Using that trial, the asynchronous procedure was simulated with the determined upper sub-trial duration.

If any p-value occurs which is lower as the determined threshold, it becomes the new threshold. To clarify, the lower the threshold, the better the accuracy and the better the non-control state detection, but contrary the longer it takes until the threshold is reached meaning longer trial durations.

Additionally, an offline analysis was performed to optimize the spelling performance using several thresholds. For this, a threshold parameter was introduced, which defines, as explained above, the percentile out of all sub-trials that lead to a misclassification. The analysis was performed using threshold parameter values between 1 and 5 (in steps of 0.1). It is worth to note that the resulting thresholds are no longer optimized for non-control state detection.

11.1.3 Hardware & Software

The same setup was used as explained in section 9.1.2. For the sake of completeness, it will be explained again.

The system consists of a g.USBamp (g.tec, Austria) EEG amplifier, two personal computers (PCs), Brainproducts Acticap system with 32 channels and an LCD monitor (BenQ XL2430-B) for stimuli presentation. Participants are seated approximately 80 cm in front of the monitor.

PC1 is used for the presentation on the LCD monitor, which is set to refresh rate of 60 Hz and its native resolution of 1920×1080 pixels. The stimuli are presented using the framework introduced in chapter 4. A stimulus can either be black or white, which can be represented by 0 or 1 in a binary sequence and is synchronized with the refresh rate of the LCD monitor. The timings of the monitor refresh cycles are synchronized with the EEG amplifier by using the parallel port.

PC2 is used for data acquisition and analysis. BCI2000 [SMH⁺04] is used as a general framework for recording the data of the EEG amplifier, and the data processing is done with MATLAB [MAT17]. The amplifier sampling rate was set to 600 Hz, resulting in 10 samples per frame/stimulus. Additionally, a TCP network connection was established to PC1 in order to send instructions to the presentation layer and to get the modulation patterns of the presented stimuli.

A 32 electrodes EEG layout was used, 30 electrodes were located at Fz, T7, C3, Cz, C4, T8, CP3, CPz, CP4, P5, P3, P1, Pz, P2, P4, P6, PO9, PO7, PO3, POz, PO4, PO8, PO10, O1, POO1, POO2, O2, OI1h, OI2h, and Iz. The remaining two electrodes were used for electrooculography (EOG), one between the eyes and one left of the left eye. The ground electrode (GND) was positioned at FCz and reference electrode (REF) at OZ.

11.1.4 Data acquisition

Initially, the participants performed a run to generate a spatial filter, and the same preprocessing steps were performed for all runs as described in section 6.1.6.

The training phase was split into three runs (32 trials each) with 4 s trial duration, and 1 s inter-trial time, those runs are used for the regression model training and threshold determination. To optimize the threshold for NC state detection, the participants performed a run with a single 2-minute trial where they have to look at the bottom. To test the asynchronous classification, the participants performed six runs (32 trials each) with 500 ms inter-trial time. To test the NC state the participants performed four runs, each starting with 0.5 min NC phase followed by 32 IC trials and an additional NC phase with 0.5 min length. During NC phases, once the monitor was covered, once the participants had to look at the bottom, once they had to look at the left of the monitor, and once to close their eyes. During all mentioned runs the matrix-keyboard layout (Fig. 4.1) was used, 32 trials each in lexicographic order.

Afterward, each participant had to perform three runs using the German QWERTZ-keyboard layout (Fig. 4.2). The participants were introduced to write "Asynchron BCI" (case-sensitive), and they should correct any errors that occur. "Asynchron" is the German word for "asynchronous". It was not prescribed how the participants should spell upper-case or lower-case letters (shift-key or caps-key).

11.1.5 Performance evaluation

The accuracy of the target prediction (Eq. 2.9) was analyzed for the lexicographic spelling as well as for the copy spelling. For the latter, the number of trials varies between different runs and participants, therefore, the number of correctly predicted trials was analyzed manually. Any trial which leads to a step closer to the desired result is defined as a correctly predicted target, which means that also correcting a miss-spelled letter is a correctly predicted target.

The corresponding ITRs were calculated using Eq. 2.10 with N the corresponding number of targets (32 or 55), P the corresponding target prediction accuracy, and T the average trial duration including the used inter-trial time of 0.5 s. Additionally, as T varies, it is also given for each participant. Furthermore, the number of correct targets (CT) per minute is given by

$$\text{CT/min} = (2 \times \text{ACC} - 1) \cdot \frac{60}{T}. \quad (11.1)$$

Since CT/min does not take case-sensitive spelling into account, also the number of correct case-sensitive letters (CL) per minute is given for the copy-spelling runs, which can be calculated by

$$\text{CL/min} = \frac{\text{total number of letters}}{\text{total number of required trials}} \cdot \frac{60}{T}. \quad (11.2)$$

As "Asynchron BCI" consists of 13 letters and the participants had to perform three runs, the total number of letters is 39. Note that the ITR cannot directly be compared to CL/min as it does not take corrections into account, if this is desired, the *Utility* measure (Eq. 2.11) should be used, as it also takes corrections into account.

11.2 Results

11.2.1 Online lexicographic spelling performance

For lexicographic spelling, a 4×8 matrix-keyboard (Fig. 4.1) was used. Table 11.1 lists the target prediction accuracy, the average trial time, the corresponding ITR and the corresponding number of correct targets (CT) per minute for each participant. The average accuracy was $99.3\% \pm 0.43\%$ with an average trial duration of $2.61 \text{ s} \pm 0.78 \text{ s}$ (including an inter-trial time of 0.5 s), which corresponds to an average ITR of $122.7 \text{ bpm} \pm 33.2 \text{ bpm}$ and $24.5 \text{ CT/min} \pm 6.7 \text{ CT/min}$, respectively. Across all participants, the minimal and maximal performance ranges from 76.2 bpm (S08) to 170.9 bpm (S02).

Table 11.1: Results using the Matrix-Layout (lexicographic).

Subject	CT/min	Accuracy [%]	ITR [bpm]	Time [s]	NC errors/min
S01	25.3	99.5	125.9	2.35	0
S02	34.2	99.0	170.9	1.72	0
S03	27.1	99.5	136.2	2.19	0
S04	15.9	100.0	80.0	3.76	0
S05	29.9	98.4	151.1	1.94	0.25
S06	19.7	99.5	99.5	3.02	0
S07	32.3	99.5	161.5	1.84	0.5
S08	15.2	99.5	76.2	3.91	0
S09	19.7	99.5	99.0	3.02	0
S10	25.5	99.0	126.9	2.31	0
mean	24.5	99.3	122.7	2.61	0.075

Shown are the results for the lexicographic spelling (matrix-layout, 32 targets, 192 trials). The number of correct targets (CT) per minute, the target prediction accuracy, the information transfer rates (ITR) and the average trial duration (including an inter trial time of 0.5 s) are given. Using the matrix-layout, several non-control (NC) states were tested (4 min in total), the results are given as the average number of erroneous classifications per minute during the NC state. Best results are in bold font.

11.2.2 Online non-control detection performance

As mentioned, the differentiation between IC and NC is important for an end-user application. Therefore, the NC states were tested under four different conditions, furthermore, both transitions were tested: IC to NC and vice versa. While each IC state was always recognized for all participants, the NC state detection resulted in 0.075 erroneous classifications per minute on average (Table 11.1). Here it is worth mentioning that 8 out of the 10 participants had a perfect NC detection with 0 erroneous classifications per minute during the NC state (which lasted 4 minutes in total).

11.2.3 Online case-sensitive copy-spelling performance

A matrix-keyboard with lexicographic order has the advantage of equal sized targets, but most participants/end-users are familiar with established keyboard layouts. Therefore, the real-case performance using a 55 target German QWERTZ-layout (Fig. 4.2) was tested by spelling three times "Asynchron BCI" (case sensitive). In case of errors, the participants had to correct them by choosing the backspace-target to delete the previous character.

Table 11.2 lists the same performance measures as for the matrix-layout, but additionally the number of correct letters (CL) per minute, which includes corrections and case-sensitive letters. The average accuracy was $98.1\% \pm 2.62\%$ with an average trial duration of $4.43 \text{ s} \pm 3.14 \text{ s}$ (including an inter-trial time of 0.5 s), which corresponds to an average ITR of $104.9 \text{ bpm} \pm 54.8 \text{ bpm}$, $18.2 \text{ CT/min} \pm 9.4 \text{ CT/min}$, and $15.6 \text{ CL/min} \pm 8.3 \text{ CL/min}$, respectively. Interestingly, compared to the matrix layout, for some of the participants, the average trial time is highly increased, especially for S01 and S08.

11.2.4 Offline threshold optimization

In general, the threshold is used to identify the correct target and to distinguish between IC and NC state. For the online experiment, the threshold was optimized for NC state detection. In order to optimize it for spelling performance, several thresholds were tested by simulating the asynchronous BCI using the lexicographic trials. Fig. 11.2 shows the accuracies, ITRs, the number of erroneous classifications during NC state and the average trial durations, averaged over all participants. The results show that the communication speed can be optimized to an ITR of 132.4 bpm with an average trial duration of 1.93 s including 0.5 s inter-trial time. However, using the corresponding threshold during the NC state results in 14.4 erroneous classifications per minute.

Table 11.2: Results using the QWERTZ-layout (case-sensitive copy-spelling).

Subject	CL/min	CT/min	Accuracy [%]	ITR [bpm]	Trials	Time [s]
S01	6.5	8.8	94.9	50.9	59	6.13
S02	29.7	34.3	100.0	198.1	45	1.75
S03	21.3	24.5	100.0	141.9	45	2.44
S04	7.2	9.1	94.2	49.5	52	6.22
S05	22.6	26.1	100.0	151.0	45	2.30
S06	11.2	12.6	97.8	72.6	46	4.55
S07	19.0	22.4	94.2	129.9	52	2.37
S08	4.3	5.0	100.0	28.7	45	12.08
S09	13.3	15.4	100.0	88.9	45	3.90
S10	20.7	23.9	100.0	137.9	45	2.52
mean	15.6	18.2	98.1	104.9	48	4.43

Shown are the results for the case-sensitive copy-spelling (German QWERTZ-layout, 55 targets, spelling 3 times "Asynchron BCI"). The number of correct targets (CT) per minute, the target prediction accuracy, the information transfer rates (ITR), the average trial duration (including an inter trial time of 0.5 s) and the number of correct letters (CL) per minute are given, whereby CL takes corrections and case-sensitive letters into account. Best results are in bold font.

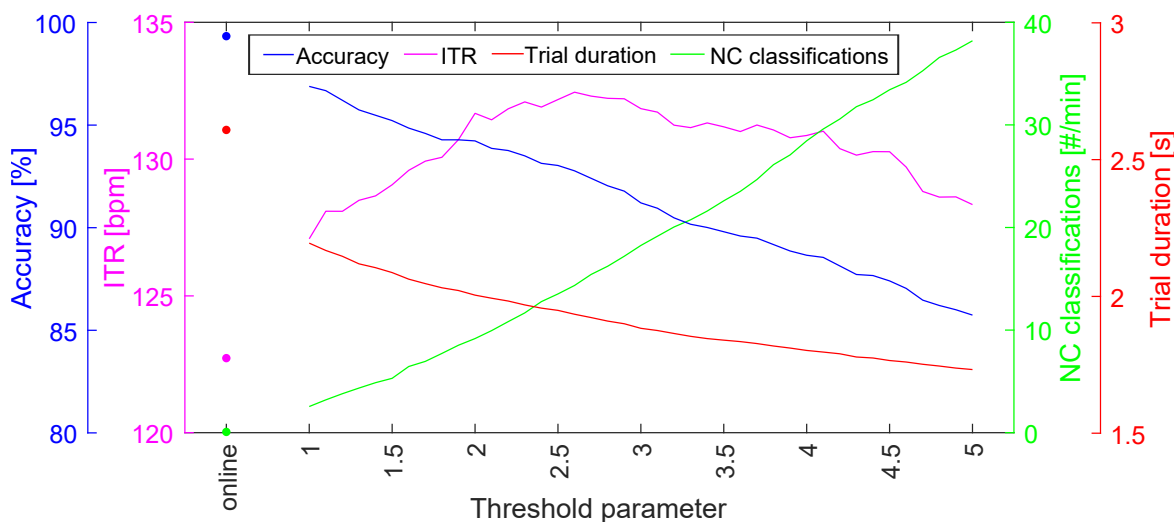


Figure 11.2: Effects of different p-value thresholds. Shown are the accuracies, information transfer rates (ITRs), erroneous classifications during the non-control state per minute and the average trial duration including an inter-trial time of 0.5 s. The results using the threshold determined during the experiment is marked as "online". The threshold parameter value defines the percentile of allowed misclassifications and the resulting threshold is the highest p-value that leads to it (see Methods for details).

11.3 Discussion

For an end-user application, the following factors must be met by a BCI: high communication speed as well as asynchrony. While the former has been addressed and achieved by several studies [SRB12b, CWN⁺15], the latter is still a key obstacle, especially for NC state detection. The presented asynchronous BCI speller achieved high communication speed (average of 122.7 bpm) as well as a robust NC state detection with only 0.075 erroneous classifications per minute during the NC state. Compared to the state-of-the-art asynchronous BCI speller [ST18] (67.7 bpm) the ITR is nearly doubled. Furthermore, among the advantage of asynchronous control, the performance is even better compared to online synchronous BCI control (chapter 9) which resulted in an ITR of 108.1 bpm.

As mentioned, besides communication speed, an end-user suitable BCI has to distinguish between intentional control (IC) and non-control (NC). Otherwise, the BCI will execute random commands during NC states, which highly reduces the user experience. While recent comparable methods for NC state detection [ZGW08, LPWY13, ABG18] achieved up to only 0.49 erroneous classifications per minute during the NC state, the presented method reduced the number of erroneous classifications by a multi-fold of factor 6.5. Here it is worth mentioning that 8 of the 10 participants had no erroneous classifications during the NC state. Since the performance of the NC detection only depends on the threshold, it can simply be optimized for a perfect NC detection. For example, instead of using only a 2 min NC run for threshold determination, the run-time could be increased in order to get the lowest p-value that can occur during an NC state. Indeed, this would reduce the spelling performance by increasing the required trial duration, but this can be counteracted by defining two thresholds: one optimized for IC state and one optimized for NC state. For example, if the user intends to go from IC to NC state a special target must be gazed, which sets the corresponding threshold optimized for NC state. Once the user intends to go back to IC state, the classification of the first intended target takes longer, but afterward, the threshold will automatically be set to the threshold optimized for IC state, which in turn allows for faster spelling. It was shown that an optimized threshold results in an increased ITR (132.4 bpm vs. 122.7 bpm). Using other methods for threshold optimization could increase the performance even more.

The spelling results revealed high variances between the participants, for example, using the QWERTZ-layout, S02 achieved 198.1 bpm while S08 achieved only 28.7 bpm. Among the physical difference that always occur between different participants, it is caused by the determination of the optimal maximal sub-trial length. For the determination, only durations between 0.5 s to 3.0 s were tested, but it turned out, that 3.0 seconds is not enough for some of the participants. For example, for S08 using the QWERTZ-layout and increasing the maximum sub-trial length to 6.0 s results in a faster classification with an average trial time of 8.1 s compared to 12.1 s using the maximum sub-trial length of 3.0 s. Fig. 11.3 shows this behavior

for a single trial. However, increasing the maximum sub-trial length has one negative effect, it will increase the first classification time after the NC state, but the advantage of faster classifications during the IC state outweigh the disadvantage.

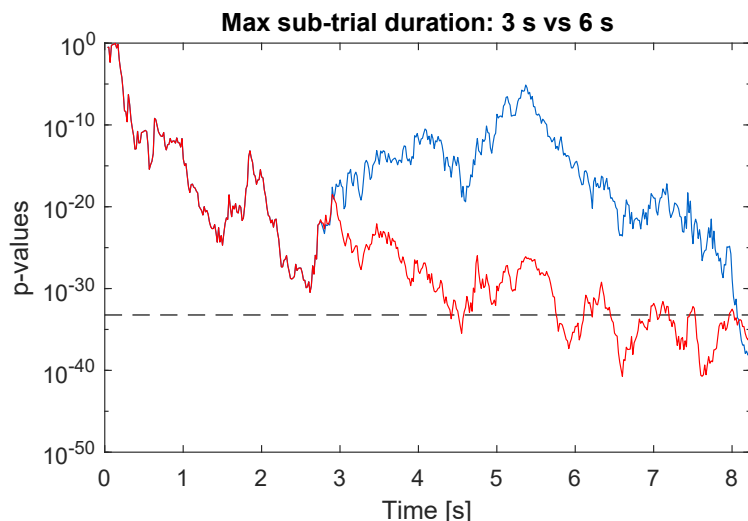


Figure 11.3: Comparison of the classification speed using different sub-trial durations. The blue line represents the corresponding p-values of the correct target of one of the trials of participant S08 using a sub-trial duration of 3 s which was determined during the online experiment, whereby the red line represents the p-values of the same trial using a sub-trial duration of 6 s. The grey dashed line indicates the p-value threshold used for S08. It clearly shows that using a sub-trial duration of 6 s results in a faster classification speed (approximately 8 s vs. 4.5 s), indicating that a maximum sub-trial length of 3 s is too short for participant S08.

Furthermore, as shown in chapter 10 the method can be used with an arbitrary number of targets without further training. This was confirmed by the current results, although trained using a 32-target matrix-keyboard, the method also works using a 55 target German QWERTZ layout. The results are slightly worse than using the matrix keyboard (104.9 bpm vs. 122.7 bpm), but this is mainly due to the reduced target size (5×5 cm vs. 3×3 cm), as this results in lower VEP responses. On the other side, 4 participants achieved an equal or even higher ITR using the QWERTZ-keyboard. Especially, S02 achieved 198.1 bpm resulting in 29.7 correct case-sensitive letters per minute. Here it is worth to note that, as mentioned, it was not prescribed how the participants should spell upper/lower-case. The optimal (fastest) solution to spell "Asynchron BCI" was "↑asynchron ↓bci" as done by most participants, but sometimes it was spelled "↓a↓synchron ↓bci" or "↑asynchron ↑b↓ci", which requires more trials. Therefore, this makes the comparability of CL/min difficult. However, apart from that, all participants have noted that a familiar keyboard layout gives a more natural spelling experience, which therefore is another important fact for an end-user application.

To summarize, with an average ITR of 122.7 bpm this is the first asynchronous high-

speed BCI. It is fully flexible with targets and has a new perfect NC state detection. As those properties are missing in other high-speed BCIs, this is the first high-speed BCI suitable for real-world applications.

Finally, as discussed in chapter 9, the stimulation, as well as the EEG2Code model training, could be designed adaptively. With the high average accuracy of 99.3% it would be possible to rate the used stimulation patterns by the resultant trial length, the shorter a trial, the more likely the corresponding stimulation pattern will be used again. Furthermore, the EEG2code model could be trained continuously, because only 0.7% would result in wrong train data.

Chapter 12

Effects of varying synchronization latencies

In the previous chapter, an asynchronous high-speed BCI with robust non-control state detection was proposed which is suitable for real-world applications. However, as explained in chapter 4 and 5, the exact synchronization of the EEG data and the presented stimulation patterns is important for the BCI performance. The synchronization was achieved by using the parallel port, however, this significantly restricts the number of devices that can be used for stimuli presentation. For real-world applications, it would be desirable to use each possible device, especially mobile devices, in order to use the system wherever the user intends to. The mobile usage is further restricted, as wired EEG electrodes/amplifier with low latencies were used in order to achieve the best synchronization.

The best-case scenario would be to use wireless electrodes in combination with a mobile device (like smartphones and tablets) for stimuli presentation and data analysis. However, this means that the synchronization can no longer be guaranteed on a millisecond scale.

This issue will be addressed in the current chapter. For this, the asynchronous BCI was simulated as described in the previous chapter, but with varying synchronization latencies between 0 ms and 50 ms. The results prove that the EEG2Code model can detect the latency itself with an average deviation of only 2.98 ms.

12.1 Methods

12.1.1 Simulate synchronization latencies

To simulate the synchronization latencies, the asynchronous BCI control was simulated as described in chapter 11, but instead of using the trial-start trigger from the

parallel port, the trial-start was varied with different latencies, which means the real trial-start trigger was shifted backward.

Depending on the used synchronization method, a maximum latency can be guaranteed. For example, if a maximum latency of 25 ms can be guaranteed, the trial-start trigger has to be shifted between 0 and 15 samples with the used sampling rate of 600 Hz. For each shift, the p-values between the EEG2Code model prediction and the corresponding stimulation patterns were calculated. If one of those p-values is lower than the p-value threshold, the target will be selected. For better understanding, this means that multiple comparisons for each stimulation pattern must be made, one for each shift. With a maximum latency of 25 ms, this corresponds to 16 comparisons between the model prediction and each stimulation pattern.

This procedure was simulated with latencies between 0 ms and 50 ms, which corresponds to a shift between 0 and 30 samples, respectively.

12.1.2 Hardware & Software

The same setup was used as explained in section 11.1.3. For the sake of completeness, it will be explained again.

The system consists of a g.USBamp (g.tec, Austria) EEG amplifier, two personal computers (PCs), Brainproducts Acticap system with 32 channels and an LCD monitor (BenQ XL2430-B) for stimuli presentation. Participants are seated approximately 80 cm in front of the monitor.

PC1 is used for the presentation on the LCD monitor, which is set to refresh rate of 60 Hz and its native resolution of 1920×1080 pixels. A stimulus can either be black or white, which can be represented by 0 or 1 in a binary sequence and is synchronized with the refresh rate of the LCD monitor. The timings of the monitor refresh cycles are synchronized with the EEG amplifier by using the parallel port.

PC2 is used for data acquisition and analysis. BCI2000 [SMH⁺04] is used as a general framework for recording the data of the EEG amplifier, and the data processing is done with MATLAB [MAT17]. The amplifier sampling rate was set to 600 Hz, resulting in 10 samples per frame/stimulus. Additionally, a TCP network connection was established to PC1 in order to send instructions to the presentation layer and to get the modulation patterns of the presented stimuli.

A 32 electrodes EEG layout was used, 30 electrodes were located at Fz, T7, C3, Cz, C4, T8, CP3, CPz, CP4, P5, P3, P1, Pz, P2, P4, P6, PO9, PO7, PO3, POz, PO4, PO8, PO10, O1, POO1, POO2, O2, OI1h, OI2h, and Iz. The remaining two electrodes were used for electrooculography (EOG), one between the eyes and one left of the left eye. The ground electrode (GND) was positioned at FCz and reference electrode (REF) at OZ.

12.1.3 Data acquisition

To test the effects of varying synchronization latencies, the asynchronous BCI control (chapter 11) was simulated using the data as described in section 11.1.4.

The participants had to perform six runs (32 trials each) in lexicographic order using the matrix-keyboard layout (Fig. 4.1) with 500 ms inter-trial time. Therefore, $6 \times 32 = 192$ trials were analyzed for each participant.

12.1.4 Performance evaluation

To analyze the performance, the target prediction accuracy is calculated by using Eq. 2.9. Furthermore, as the real trial-start trigger is known (through the parallel port), also the deviation to the detected trial-start was determined. The detected trial-start is the corresponding shift which had the lowest p-value and led to a target classification (is smaller than the corresponding threshold).

12.2 Results

Fig. 12.1 shows the results averaged over all participants. Shown are the target classification accuracies as well as the average deviations of the detected latencies to the real trial-starts in respect to the simulated maximum latency. The results using a maximum latency of 0 ms represents the online BCI performance of the previous chapter using the exact trial start trigger of the parallel port. If a maximum latency of 5 ms can be guaranteed, the classification accuracy drops by 0.8% to 98.4% compared to the online results, whereby the real trial-start can be detected with an

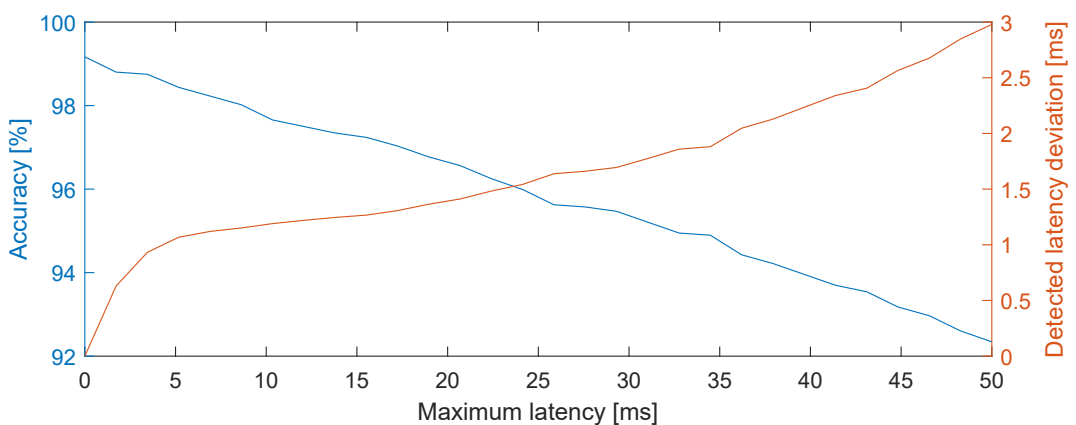


Figure 12.1: Effects of varying synchronization latencies. Shown are the target prediction accuracies as well as the deviation between the detected trial-start and the real trial-start. Both in respect to the simulated maximum latency.

average deviation of 1.07 ms. Using maximum latencies between 5 ms and 50 ms results in a nearly linear decrease of the classification accuracy with the lowest accuracy of 92.3% using a maximum latency of 50 ms, whereas the deviation of the detected trial-start to the real trial-start increases to 2.98 ms, also approximately linear. The average trial durations are not shown, as they remain approximately the same as during the online experiment (2.5 s vs. 2.6 s), which is as expected.

12.3 Discussion

To evaluate the effects of varying synchronization latencies, several maximum latencies were tested, whereas, in a real-world scenario, those latencies can be caused due to a data transfer of wireless EEG electrodes/amplifiers, as well as by using a non-static method to synchronize the stimuli presentation with the EEG data.

During the previous experiments, the synchronization was done by using the parallel port, which has low latencies in the range of 10^{-9} s. Alternatively, the synchronization could be done by using time-stamps, which in turn requires to synchronize the time between the hardware used for stimuli presentation and the hardware/software used for EEG recording/analysis. If different hardware is used for both layers, like done during the previous experiments, the time-stamps could be sent, for example, using TCP/IP.

Exemplary, if a maximum latency of 50 ms can be guaranteed between the measured EEG data and the stimuli presentation, the results show that the performance drops to 92.3%. It is assumed, that a maximum latency of 10 ms should be achievable for time synchronization, which would result in an accuracy of approximately 98%. Regardless of this, since a classification depends on the used p-value threshold, the worse performance can be counteracted by optimizing the threshold in order to reduce the miss-classifications.

It is worth to note that the EEG2Code model was able to determine the real trial-start with a deviation of only 2.98 ms using a maximum latency of 50 ms, which corresponds to an average shift of 1.79 samples within the possible shift of 30 samples. With a maximum latency of 10 ms, the deviation is only approximately 1 ms which is less than a single sample. On the one hand, this proves that the synchronization of the measured EEG with the stimuli presentation is indeed within a millisecond scale, and on the other hand it proves that the EEG2Code model can detect the latency itself with a small deviation. It must be noted, that the data acquisition was performed using the framework described in chapter 4, which synchronizes each drawn frame/stimulus with the EEG. This means that a possibly occurring frame drop, as well as small variation in the monitor's refresh rate, can easily be corrected by mapping the stimulation patterns to the EEG in respect to each drawn frame. Using the approach proposed in this chapter would also require to use, for example,

time stamps to address this.

The results reveal that the proposed asynchronous BCI speller can also be used without an exact synchronization. This offers the possibility to use it with mobile hardware with which an exact synchronization cannot be realized. For example, with a wireless EEG amplifier in combination with a tablet/smartphone for stimuli presentation and data analysis. This, in turn, allows using the system, for example, on-the-way and therefore it further improves the usability for real-world applications.

Chapter 13

Improve performance using deep-learning

In chapter 6 the EEG2Code model was introduced by using a linear machine learning approach, the ridge regression. The method was used as it is fast to calculate and linearity of VEP generation was proposed in previous works [CPAD⁺11, LPR⁺06]. However, the results of chapter 7 suggest that non-linear processes take place during the VEP generation.

There are several non-linear machine learning approaches, like support vector machines with non-linear kernels [BGV92], the non-linear regression approach by Seber *et al.* [SW03], as well as deep-learning approaches. Of these, the latter has shown to improve the performance compared to "traditional" machine learning approaches, especially for image classification and speech recognition [SZ14].

One class of deep-learning algorithms are convolutional neural networks (CNNs) (see section 2.5.2), which were already used in the field of BCIs. The first work which explored CNNs for BCI is by Cecotti *et al.* [CG11]. They used 4-layer topology to classify P300 ERPs, whereby the first layer was designed to learn spatial filters, the second to learn temporal filters, and the third/fourth are fully connected layers which perform the classification (P300 or not). They used the P300-speller data from the third BCI competition [BMK⁺06] to evaluate their CNN and outperformed the methods used by the BCI competition winners. However, since the data set contains only data of two subjects, no significant statement can be made.

Recently, Kwak *et al.* proposed a 3-layer CNN that uses frequency features as input for robust SSVEP detection [KML17]. They compared their CNN approach to other state-of-the-art methods for SSVEP decoding and outperformed all of them. Especially for noisy EEG data obtained by a moving participant they achieved an accuracy of 94.03%, which was significantly better than all other methods. Thomas *et al.* performed a similar comparison, with the result that the CNN outperformed all other state-of-the-art methods as well [TMS⁺17].

In addition to the findings of chapter 7, those results suggest using a non-linear deep learning approach. In this chapter also a CNN is used, as it is expected to significantly improve the performance of the EEG2Code model.

Parts of the results, figures and wording are already published in [NS19b].

13.1 Methods

13.1.1 Convolutional EEG2Code model

The general topology of the CNN model is from the master thesis of Felix van Gunsteren [Gun18] and was modified for the current work. A detailed structure of the EEG2Code CNN is depicted in Fig. 13.1 which also shows the input/output shape of each operation. A detailed explanation of each operation can be found in section 2.5.2. Contrary to the EEG2Code ridge regression model, the data is not spatially filtered beforehand. This means the model takes windows of size $T \times C$, whereby $T = 150$ corresponds to the number of time samples (250 ms), and $C = 32$ corresponds to the number of EEG channels. The model consists of five layers, whereby the first layer uses convolutional kernels of size 1×32 , which means they move over the channels and act as spatial filters. As a 16 Conv operation is performed, this corresponds to 16 spatial filters in total. The idea was adapted from Lawhern *et al.* [LSW⁺18] although they did not use it during the first layer. Contrary, the second and third layer act as temporal filters and move over the time samples. As mentioned by van Gunsteren, the first two layers could be combined in a single layer with kernel size (64×32) , but this would increase the number of parameters: $(64 \cdot 32 + 1) \cdot 16 = 32784$ operations compared to $(1 \cdot 32 + 1) \cdot 16 + (64 \cdot 1 + 1) \cdot 16 = 1568$ operations. Furthermore, as shown by Lawhern *et al.* separating those layers performs well on EEG data [LSW⁺18].

MaxPooling operations are performed to reduce the size of intermediate representation between layers. The BatchNorm operation results in a faster and more stable training. The Dropout operation is used for regularization to avoid over-fitting of the model. The Dense operation fully connects the input neurons with the given number of output neurons. The Activation operations define/transform the output of each neuron depending on a given activation function, whereby the *softmax* function takes an un-normalized vector, and normalizes it into a probability distribution, and therefore gives the probabilities that the initial input belongs to one of the classes. Since the model is trained on two classes (0 or 1), the outputs are p_0 and p_1 , the probabilities for each class. It must be noted that $p_1 = 1 - p_0$. The model is trained using a learning rate of 0.001 and a batch size of 256. In total 25 epochs were trained, and the best model was selected as the final model.

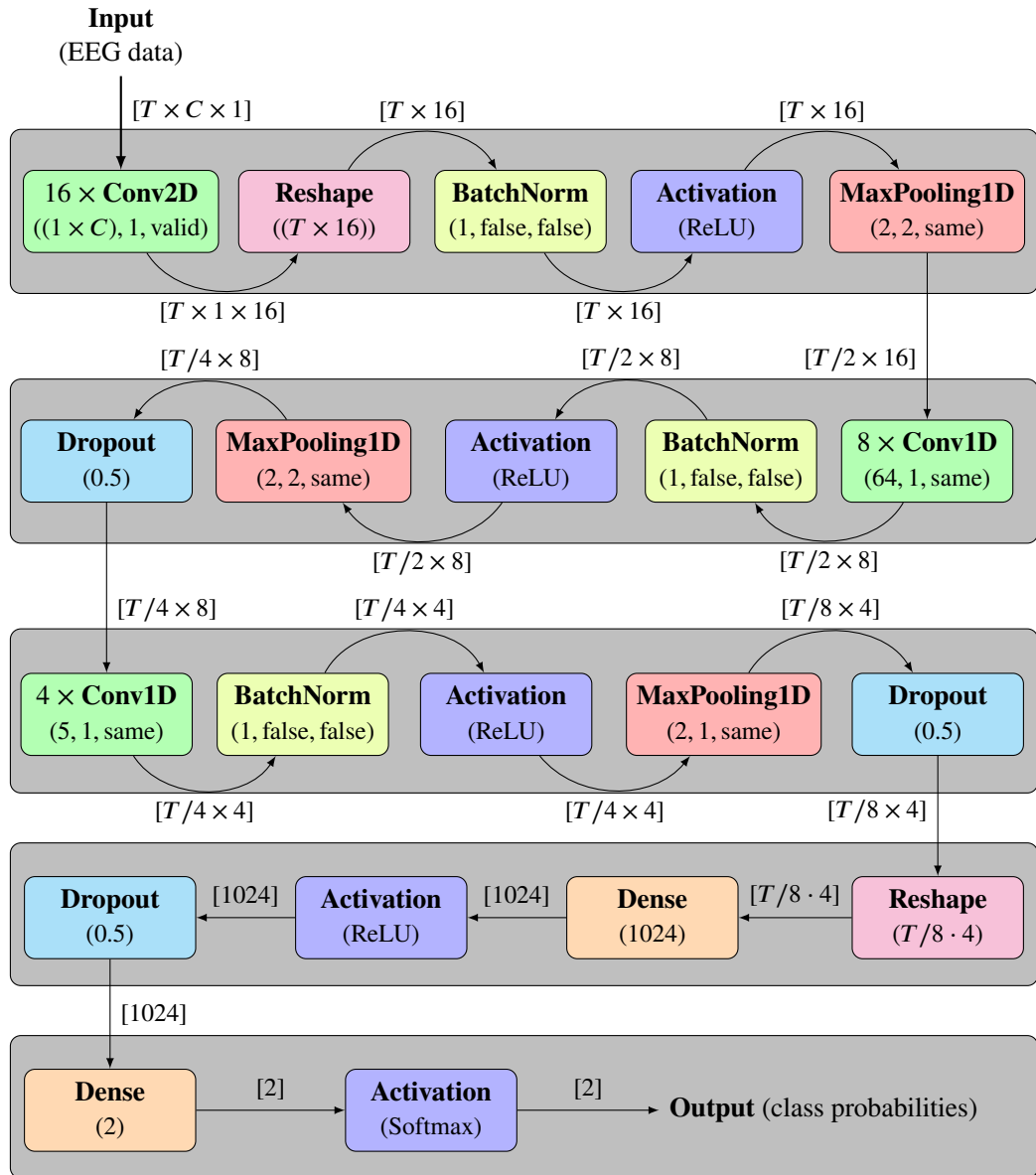


Figure 13.1: Topology of the EEG2Code convolutional neural network. The five layers are indicated by the gray boxes. Each layer consists of different operations, whereby the same operations are colored the same. Below each operation, the used parameters are given, whereby the parameters for the Conv and MaxPooling are (kernel/pooling size, stride, padding), and for BatchNorm (axis, scale, center). The edges are labeled with the shape of the input and output data, respectively. The input of the model is a 250 ms ($T = 150$ samples) window of EEG data with $C = 32$ channels. The output is the probability that the input belongs to one of the classes (binary 1 or 0). As the window is shifted sample-wise over the complete trial data, the output is the prediction of the corresponding stimulation pattern. The general topology was proposed by Felix van Gunsteren [Gun18] for P300 classification and was modified for the current work.

13.1.2 Hardware & Software

A similar setup was used as for the (a)synchronous BCI control (see section 9.1.2), except that the EEG2Code CNN computations were performed on an IBM Power System S822LC with four Nvidia[®] Tesla P100 GPUs using Python v2.7 [Pyt10] and the Keras framework [C⁺15]. The server was provided by the *IBM Shared University Research Grant* and includes the IBM PowerAI environment.

The system consists of a g.USBamp (g.tec, Austria) EEG amplifier, three personal computers (PCs), Brainproducts Acticap system with 32 channels and an LCD monitor (BenQ XL2430-B) for stimuli presentation. Participants are seated approximately 80 cm in front of the monitor.

PC1 is used for the presentation on the LCD monitor, which is set to refresh rate of 60 Hz and its native resolution of 1920×1080 pixels. A stimulus can either be black or white, which can be represented by 0 or 1 in a binary sequence and is synchronized with the refresh rate of the LCD monitor. The timings of the monitor refresh cycles are synchronized with the EEG amplifier by using the parallel port.

PC2 is used for data acquisition, whereby BCI2000 [SMH⁺04] is used as a general framework for recording the data of the EEG amplifier. The amplifier sampling rate was set to 600 Hz, resulting in 10 samples per frame/stimulus. A TCP network connection was established to PC1 in order to send instructions to the presentation layer and to get the modulation patterns of the presented stimuli. During the online experiment, the EEG data was continuously sent to PC3 using a TCP connection. PC3 performs the EEG2Code prediction and sent the classification to PC2.

A 32 electrodes EEG layout was used, 30 electrodes were located at Fz, T7, C3, Cz, C4, T8, CP3, CPz, CP4, P5, P3, P1, Pz, P2, P4, P6, PO9, PO7, PO3, POz, PO4, PO8, PO10, O1, POO1, POO2, O2, OI1h, OI2h, and Iz. The remaining two electrodes were used for electrooculography (EOG), one between the eyes and one left of the left eye. The ground electrode (GND) was positioned at FCz and reference electrode (REF) at OZ.

13.1.3 Data acquisition

It shall be shown that the proposed EEG2code CNN model results in better performances compared to the EEG2code ridge regression model (see section 6). Therefore, an offline analysis was performed using the data of the previous chapters, whereby the data of the training runs was split into two equal-sized sub-sets, one used as training data and one used as validation data.

For the bit-prediction accuracy, as well as for the synchronous BCI control, the data from chapter 9 is used. The important parts will be explained again. The training phase was split into three runs, but with varying trial duration, 5 s, 4 s, and 3 s, re-

spectively. During the training, fully random stimulation patterns (see section 6.1.3) were used. The testing phase was split into 14 runs with a trial duration of 2 s. The testing runs were alternated using fully random stimulation patterns and optimized stimulation patterns (see section 8.1.3). For the bit-prediction accuracy, the seven runs with fully random stimulation were used, whereby for the synchronous BCI control, the seven runs with optimized stimulation were used. The participants had to perform each run in lexicographic order, with 32 targets per run. It should be mentioned that, in case of the bit-prediction, a "trial" is actually a single 250 ms window.

For the asynchronous BCI control, the data from chapter 11 is used. The necessary parts are explained again. The training phase was split into three runs with 4 s trial duration and 1 s inter-trial time, whereby the optimized stimulation was used. The testing phase consists of six runs with 500 ms inter-trial time, also 32 trials in lexicographic order with optimized stimulation. For comparison reasons, the same p-value threshold was used as determined during the online experiment (see section 11.1.2).

Furthermore, subject S01 who participated in the synchronous BCI control experiment, also performed an additional experiment to prove that the bit-prediction performance can also be achieved during an online analysis. For this, the participant first performed the same training phase as for the asynchronous BCI control. Afterward, 96 runs were performed with 5 s of random stimulation each, which means 300 random bits were presented per run.

13.1.4 Performance evaluation

Since the analysis includes the EEG2code pattern prediction, the synchronous BCI control, as well as the asynchronous BCI control, the same performance measures were used as described in sections 6.1.7, 9.1.5, and 11.1.5.

13.2 Results

13.2.1 Offline analysis: Simulated online experiment

First, the EEG2Code stimulation pattern prediction was analyzed which resulted in an average accuracy of 74.9% using the fully random stimulation patterns, which corresponds to an average ITR of 701.3 bit/min. It is worth to note that for S01 an average accuracy of 83.4% was achieved, which corresponds to 1262.1 bit/min. Additionally, the synchronous BCI control was simulated, which results in an average accuracy of 99.5% using the optimized stimulation patterns and a trial duration of 2 s, which was used during the online BCI experiment. As the offline analysis in chapter 10 suggest that a trial duration of 1 s results in the best BCI performance,

Table 13.1: EEG2Code and synchronous BCI performance.

Subject	Pattern prediction		BCI control (2 s)		BCI control (1 s)	
	ACC [%]	ITR [bpm]	ACC [%]	ITR [bpm]	ACC [%]	ITR [bpm]
S01	83.4	1262.1	100.0	109.1	100.0	200.0
S02	72.9	567.3	99.6	107.7	94.6	177.3
S03	72.5	545.7	100.0	109.1	95.5	180.6
S04	76.8	787.9	100.0	109.1	98.7	193.2
S05	78.7	908.9	100.0	109.1	99.6	197.5
S06	75.4	704.9	100.0	109.1	98.7	193.2
S07	68.5	363.9	98.7	105.4	87.9	154.9
S08	77.1	807.1	100.0	109.1	99.6	197.5
S09	68.5	363.7	97.3	102.3	87.5	153.5
mean	74.9	701.3	99.5	107.8	95.9	183.1

Shown are the average results of all subjects, whereby best results are in bold font. The EEG2Code pattern prediction as well as the results for the simulated synchronous BCI control are shown. For both, the accuracy (ACC) as well as the ITR are given. The ITRs are calculated using Eq. 2.10 with $N = 2$ ($N = 32$) and $T = 1/60$ s ($T = 2.75$ s, $T = 1.5$ s).

the BCI control was also simulated using a trial duration of 1 s. This results in an average accuracy of 95.9% which corresponds to an ITR of 183.1 bit/min including the inter-trial time of 0.5 s. Detailed results are listed in Table 13.1.

Furthermore, also the asynchronous BCI control was simulated using the 32-target matrix-keyboard layout (Fig 4.1). The results for each participant are shown in Table 13.2. The average target prediction accuracy is 98.5% (ITR: 175.5 bit/min) with an average trial duration of 1.71 s (including 0.5 s inter-trial time). In total, 91.6% of all trials could be classified faster compared to the ridge regression model. Finally, this results in an average spelling speed of 35.3 correct targets per minute with a maximum of 48.2 CT/min.

13.2.2 Online experiment

As discussed in chapter 6, the ITR calculation of the bit-prediction with $T = 1/60$ s is slightly biased due to the required window length of 250 ms. Therefore, and to show that the bit-prediction performance can also be achieved in an online analysis, a single-subject experiment was performed with subject S01 as a proof-of-concept. In average for all 5 s runs, a bit-prediction accuracy of 83.4% was achieved which corresponds to an average ITR of 1236.7 bit/min using an unbiased $T = 5/285$ s. The results for each run are depicted in Fig. 13.2.

Table 13.2: Asynchronous spelling performance using the matrix-layout.

Subject	CT/min	Accuracy [%]	ITR [bpm]	Time [s]	Faster [%]
A01	33.3	99.0	165.4	1.77	87.0
A02	48.2	97.9	238.9	1.19	97.4
A03	39.7	99.5	197.8	1.49	91.7
A04	28.7	99.5	142.8	2.07	91.1
A05	35.8	95.3	177.8	1.52	87.0
A06	30.7	99.0	152.4	1.92	93.2
A07	45.1	100.0	225.3	1.33	96.9
A08	25.8	97.4	127.8	2.21	90.1
A09	30.9	100.0	154.4	1.94	91.1
A10	34.8	97.4	172.3	1.64	90.1
mean	35.3	98.5	175.5	1.71	91.6

Shown are the results for the lexicographic-spelling (matrix-layout, 32 targets). The number of correct targets (CT) per minute, the target prediction accuracy, the information transfer rates (ITR), the average trial duration (including an inter trial time of 0.5 s) and the percentage of trials which could be predicted faster compared to the ridge regression model. Best results are in bold font. The abbreviations of the subjects were renamed to avoid confusion.

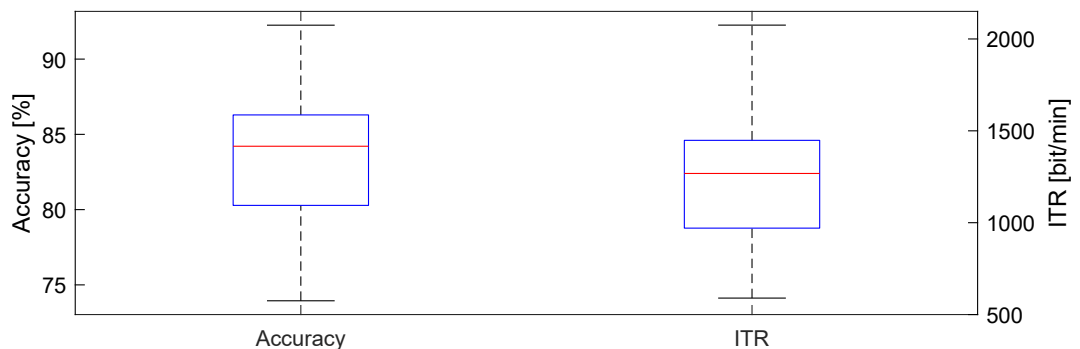


Figure 13.2: Bit-prediction accuracies and corresponding ITRs achieved during an on-line experiment of subject S01. The performances of all 96 runs, with 5 s stimulation time each, are shown. During each run 300 bits were presented, whereby only 285 could be predicted due to the window size of 250 ms (15 bits). The ITRs were calculated using Eq. 2.10 with $N = 2$ and $T = 5/285$ s.

13.3 Discussion

The results of chapter 7 suggest that non-linear processes take place during the VEP generation. Since the EEG2Code model, introduced in chapter 6, was based on the linear ridge regression approach, it was expected to achieve a better performance by using a non-linear approach.

By using the multi-layer convolutional neural network, introduced in this chapter, an average bit-prediction accuracy of 74.9% was achieved which corresponds to an ITR of 701.3 bit/min, therefore, it significantly ($p < 0.05$, paired t-test) outperformed the EEG2Code model based on ridge regression (232.0 bit/min), with more than tripling the ITR. Noteworthy, S01 achieved an average ITR of 1.26 Kbit/min, which is an outstanding performance for a passive BCI. Interestingly, S01 achieved exactly the same bit-prediction accuracy of 83.4% during the online analysis. Here it is worth to mention that the "worst" 5 s run resulted in an accuracy of 73.9 %, whereas the best run resulted in 92.3%. The latter means that 263 out of the 285 random bits could be correctly predicted, which corresponds to an ITR of 2.08 Kbit/min.

Compared to current state-of-the-art approaches, the EEG2Code CNN approach clearly outperforms the previously fastest system by Chen et al. [CWN⁺15]. They reported the previously highest ITR for a BCI, with an average ITR of 267 bit/min and an online ITR of 319 bit/min for the best subject, which was raised with the current approach to 701 bit/min and 1237 bit/min, respectively.

The performance improvement can also be confirmed for the synchronous as well as the asynchronous BCI control. For the former, the ITR could be improved by 18.6% using a trial duration of 1 s. For the latter, although the average accuracy is reduced by 0.8%, the average ITR has increased by 43.0% to 175.5 bit/min, which is due to the reduced average trial time (1.71 s vs. 2.61 s). Furthermore, with 35.3 correct targets per minute, the spelling speed could be improved by 45.3%. Here it must be noted, that the asynchronous performance is slightly biased, in average 91.6% of all trials could be predicted faster, whereas the remaining 8.4% would require a longer trial duration, which obviously cannot be simulated. Therefore, the average trial duration of 1.71 s would actually be slightly longer.

Especially the worse participants benefit from the deep learning approach, for example, the asynchronous spelling ITR of subject A08 has increased from 76.2 bit/min to 127.8 bit/min, which represents the lower limit for the asynchronous BCI speller. Furthermore, even the best performing participant A02 has profited, and the ITR has increased by 68.0 bit/min to 238.9 bit/min, which represents the upper limit and shows the potential of the presented asynchronous BCI.

Comparing the bit-prediction performance and the spelling performance, for the latter a ceiling effect can be observed. The visual information that can be decoded from the EEG is substantially higher than the information that can be achieved for spelling. While tripling the former, the latter is only increased by 43%. This can be explained by the fact that the latter depends, among other facts, on the reaction time of the subject, which is limited. For instance, to achieve a spelling ITR of 700 bit/min with a 32-target layout, the trial duration (including inter-trial time) must be reduced to 430 ms with an accuracy of 100%. This means, a subject has to spell a letter and to find the subsequent letter within 430 ms, which is nearly impossible, at least for the long term. Furthermore, as the EEG2Code is designed to use 250 ms

(150 samples) windows, this is a further limitation. For example, a stimulation time of 300 ms (180 samples) results in only 30 predicted samples, which corresponds to 3 bits of the stimulation pattern and is obviously not enough for target classification, at least for a multi-target layout.

To summarize, deep learning approaches have shown to improve the classification compared to traditional machine learning approaches for BCI control as well as in other research areas [SZ14, CG11, BMK⁺06, KML17, TMS⁺17]. In this chapter, it was confirmed that a non-linear deep-learning approach also significantly improves the BCI performance, at least for the EEG2code approach. With an asynchronous spelling performance of 175.5 bit/min, the approach is approximately three times faster compared to the previous state-of-the-art approach by Suefusa *et al.* which achieved 67.7 bit/min [ST18].

Finally, due to the observed ceiling effect, the results suggest that more powerful approaches for brain signal decoding will not translate into substantially better VEP BCI control anymore.

Chapter 14

Reduced training data and cross-subject performance

In the previous chapters, several advantages and optimizations of the EEG2Code model were proposed, which improve the end-user suitability. However, one important point has not been addressed: the amount of required training data. Collecting training data takes time and is inconvenient for the user, therefore, the time required to collect the training data should be as short as possible.

Generally, there are several ways to implement this. The most obvious is to test how the BCI performance varies with a different amount of training data and to choose the best trade-off between performance and training time. Another option is to avoid training at all. As already explained, for SSVEP BCIs it is possible to detect the corresponding frequencies without prior training, but recent works have shown a relatively poor performance [FVG07, CC14, ZXC14]. Contrary to use a train-free method, it is possible to perform a subject-independent training, which means the training is done using the EEG data of other users [LGZ08a, LGZ08b, RFPN11]. Those methods show an increased performance compared to train-free methods, but not compared subject-dependent methods. A further approach is to use not only subject-independent data but to further optimized the resultant model with subject-dependent data, which has shown to further increase the performance [YCW⁺15, WFG⁺16, CWNJ18].

In the previous chapters, the amount of training data used to train the EEG2Code model has been arbitrarily determined during the experimental design, as it was assumed that approximately 6 minutes of random stimulation should be sufficient to cover most of the possible VEP responses. In this chapter, the performance of the EEG2Code CNN model will be analyzed with respect to a varying amount of training data. Furthermore, a subject-independent classification is performed, whereby for each subject a model will be trained only on the data of other subjects.

14.1 Methods

14.1.1 Reduce training data

The amount of required training data used to train the EEG2Code model has been arbitrarily determined during the experimental design and resulted in $3 \cdot 32 \cdot 4 = 384$ s of training data. With the used inter-trial time of 1 s, this results in a total training duration of 8 min.

To test the performance with reduced training data, an offline analysis is performed by varying the amount of training data between 10% and 100% in steps of 10%. For example, a train size of 10% corresponds to approximately 38 s of training data. For training, the EEG2Code CNN model, proposed in chapter 13, is used with the same parameters. Also, the corresponding amount of training data is split into two sets of equal size, one used for training and one used for validation, respectively.

The models are used to analyze the bit prediction accuracy as well as the synchronous BCI control performance using trials with 1 s trial duration.

14.1.2 Cross-subject

Instead of reducing the train duration, it would be best if no training is required at all, therefore, an offline cross-subject analysis is performed. Again, the EEG2Code CNN model (chapter 13) is used with the same parameters. For each participant, a model is trained on the training data of 5 participants and validated on the training data of 3 participants. In the following, the model is used to analyze the bit prediction accuracy as well as the synchronous BCI control performance using trials with 2 s trial duration.

14.1.3 Hardware & Software

The same setup was used as described in the previous chapter (see section 13.1.2). For the sake of completeness, it will be explained again.

The system consists of a g.USBamp (g.tec, Austria) EEG amplifier, two personal computers (PCs), Brainproducts Acticap system with 32 channels and an LCD monitor (BenQ XL2430-B) for stimuli presentation. Participants are seated approximately 80 cm in front of the monitor.

PC1 is used for the presentation on the LCD monitor, which is set to refresh rate of 60 Hz and its native resolution of 1920×1080 pixels. A stimulus can either be black or white, which can be represented by 0 or 1 in a binary sequence and is synchronized with the refresh rate of the LCD monitor. The timings of the monitor

refresh cycles are synchronized with the EEG amplifier by using the parallel port.

PC2 is used for data acquisition, whereby BCI2000 [SMH⁺04] is used as a general framework for recording the data of the EEG amplifier. The amplifier sampling rate was set to 600 Hz, resulting in 10 samples per frame/stimulus. Additionally, a TCP network connection was established to PC1 in order to send instructions to the presentation layer and to get the modulation patterns of the presented stimuli.

A 32 electrodes EEG layout was used, 30 electrodes were located at Fz, T7, C3, Cz, C4, T8, CP3, CPz, CP4, P5, P3, P1, Pz, P2, P4, P6, PO9, PO7, PO3, POz, PO4, PO8, PO10, O1, POO1, POO2, O2, OI1h, OI2h, and Iz. The remaining two electrodes were used for electrooculography (EOG), one between the eyes and one left of the left eye. The ground electrode (GND) was positioned at FCz and reference electrode (REF) at OZ.

The data was analyzed on an IBM Power System S822LC with four Nvidia[®] Tesla P100 GPUs using Python v2.7 [Pyt10] and the Keras framework [C⁺15]. The server was provided by the *IBM Shared University Research Grant* and includes the IBM PowerAI environment.

14.1.4 Data acquisition

To analyze the performance of the EEG2code CNN model under the mentioned conditions, the data of the synchronous BCI control (section 9.1.3) is used. Since a worse performance is expected, compared to using the complete training data, an offline analysis of the asynchronous BCI control makes no sense, as the threshold will most likely not be achieved.

The complete training data of each participant consists of runs three runs with 32 trials each. The trial durations of those runs are 5 s, 4 s, and 3 s, respectively, whereby the targets were modulated with fully random stimulation patterns.

To test the bit prediction accuracy, the seven runs (32 trials each, 2 s trial duration) with fully random stimulation are used. For the synchronous BCI control analysis, the seven runs (32 trials each, 2 s trial duration) with optimized stimulation are used. The participants had to perform all runs in lexicographic order using the matrix keyboard layout (Fig. 4.1).

14.1.5 Performance evaluation

Since the analysis includes the EEG2code pattern prediction and the synchronous BCI control, the same performance measures were used as described in sections 6.1.7 and 9.1.5.

14.2 Results

Fig. 14.1 shows the bit prediction accuracies for each participant (colored lines) as well as the average accuracy (black thick line) depending on the used train size, whereby 100% corresponds to 384 s of training data. As expected, the accuracy decreases with the decreasing amount of training data. With 100% of the training data, the average accuracy is 71.2% (ITR: 504.5 bit/min) and decreases to 67.4% (ITR: 335.7 bit/min) with 10% train size (38 s). Interestingly, the loss of information is substantially greater for S01 (loss: -405.8 bit/min) compared to the other participants (average loss: -139.2 bit/min, SD: 50.1 bit/min). On the other side, S01 achieved 74.7% using 10% of training data, which is even more as all other participants achieved with 100% of training data. Although, the accuracy increase is flattening out, a significant ($p < 0.05$, paired t-test) increase was always found by doubling the train size.

The findings can be confirmed for the synchronous BCI control with 1 s trial duration. The results are depicted in Fig. 14.2. The average accuracy drops from 95.9% to 89.4% by using only 10% of the training data. Although, S01 shows the largest loss of information for the bit prediction accuracy, the BCI control accuracy of S09 drops substantially more (-24.1%) compared to the other participants (average loss: -4.2%, SD: 3.1%). Contrary to the bit prediction accuracy, a significant ($p < 0.05$, paired t-test) increase of the accuracy can only be found by doubling the train size from 10% to 20%, from 40% to 80%, and from 50% to 100%, respectively.

The results of the cross-subject analysis can be found in Table 14.1. The analysis of the bit prediction accuracy reveals an average of 59.1% and up to 61.1%. For the

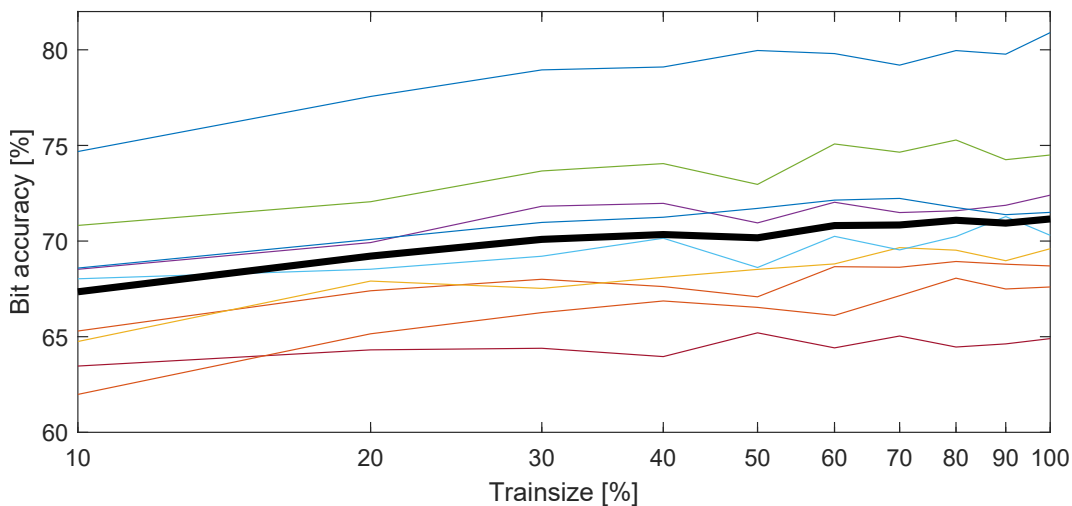


Figure 14.1: Bit prediction performance of the EEG2Code CNN model using different amount of training data. Each colored thin line represents a participant and the black thick line is average over all participants. The train size has a logarithmic scale.

synchronous BCI control, an average accuracy of 62.9% was achieved. Here it is worth to note that S04 achieved the best accuracy of 82.1% which corresponds to an ITR of 75.0 bit/min.

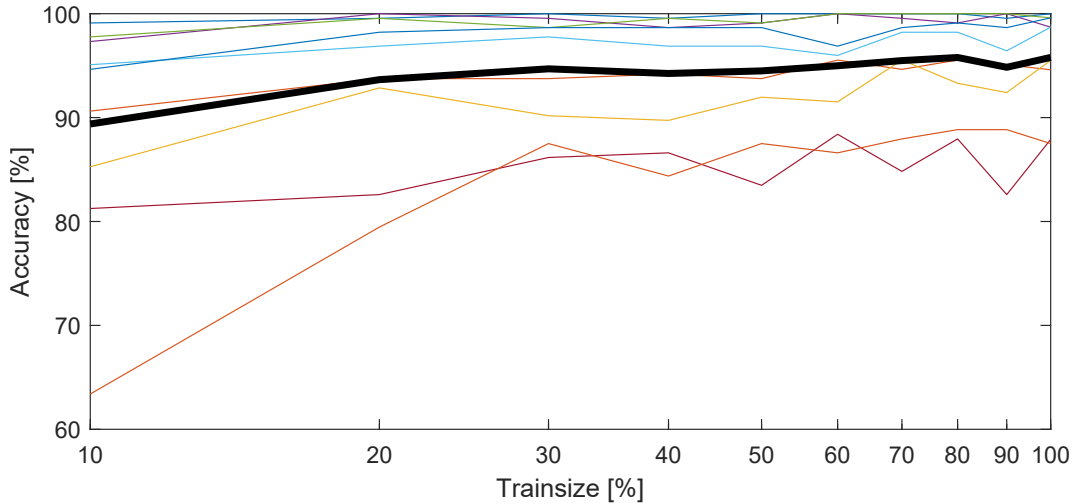


Figure 14.2: Synchronous BCI control performance of the EEG2Code CNN model using different amount of training data. Used are trials of 1 s length. Each colored thin line represents a participant and the black thick line is average over all participants. The train size has a logarithmic scale.

Table 14.1: Cross-Subject results.

Subject	Pattern prediction		BCI control	
	ACC [%]	ITR [bpm]	ACC [%]	ITR [bpm]
S01	61.1	128.6	73.7	62.5
S02	57.7	61.2	52.2	35.7
S03	58.8	80.8	60.3	45.0
S04	60.3	110.4	82.1	75.0
S05	60.2	109.4	65.6	51.7
S06	59.2	87.8	57.6	41.8
S07	58.9	83.1	61.2	46.1
S08	56.6	45.2	42.0	24.9
S09	59.0	83.8	71.4	59.4
mean	59.1	87.8	62.9	49.1
mean (within)	74.9	701.3	99.5	107.8

Shown are the average results of all subjects, whereby best results are in bold font. The left part shows the results for the EEG2Code pattern prediction, whereas the right part shows the results for the simulated synchronous BCI control. For both, the accuracy (ACC) as well as the ITR are given. The ITRs are calculated using Eq. 2.10 with $N = 2$ ($N = 32$) and $T = 1/60s$ ($T = 2.75s$). For comparison, the within-subject results are shown in the last row.

14.3 Discussion

Although the entire train duration during the previous experiments lasted for only 8 min, it is desired to reduce the train duration to a minimum, as collecting training data is inconvenient for the user. For comparison, the state-of-the-art synchronous BCI speller by Chen *et al.* required exactly the same train duration [CWN⁺15].

While the train duration was arbitrarily determined for the previous experiments, a detailed performance analysis of the EEG2Code CNN model was performed by reducing the training data. Interestingly, using only 10% of the training data, which corresponds to only 38 s of stimulation, reduced the average synchronous BCI control accuracy by only 6.5%. When using 30% of the training data, this results in an accuracy loss of less than 1%, which suggests reducing the training time to approximately 2 min. Contrary, the results also revealed that doubling the training data always results in a significant improvement, which further suggests increasing the train duration to get the best possible performance. It would also be interesting to know at which training duration the performance cannot be further improved.

While reducing the training data resulted only in a moderate performance loss, the cross-subject analysis resulted in an ITR of only 49.1 bit/min, but this was expected due to the inter-subject variability of VEP responses. Furthermore, the model performs spatial filtering, which depends on the placement of the EEG electrodes. Although the placement was standardized, the position of the EEG cap on the scalp can only be guaranteed to a certain degree. Additionally, the shape of the head varies for different users. Nevertheless, the results are in the range of the best train-free approaches [FVG07, CC14].

Finally, although a subject-independent model would be the most user-friendly approach, the results revealed that even a training phase of approximately 40 s resulted in a highly increased performance: 49.1 bit/min vs. 136.7 bit/min using the same inter-trial time. Obviously, due to the improved performance, one would take the training phase in the acceptance.

Chapter 15

Peripheral perception of different colors

The recently presented BCI, as well as most other VEP-based BCIs, make use of alternating white-black stimuli [CWN⁺15, BGW⁺11, SRB12a]. However, also chromatic stimuli result in VEP responses and previous studies have shown that achromatic and different chromatic stimulation results in different VEPs with different amplitudes as well as with different temporal courses [Reg66, KMP87, KRM96, RSC⁺94, MRMK96]. Several chromatic stimuli (e.g., red-black, green-black, blue-black) were tested by Duszyk *et al.* [DBR⁺14] with an SSVEP stimulation paradigm and by Wei *et al.* [WFL16] with a cVEP stimulation paradigm. Both studies found the best performance using white-black stimuli. Interestingly, Yan *et al.* have shown that low-frequency red-green stimuli result in a better frequency recognition [YXL⁺17], which was also confirmed by Nezamfar *et al.* for a cVEP BCI which achieved a better accuracy by using red-green stimuli [NSE15]. Aminaka *et al.* compared blue-green stimuli to white-black stimuli with a cVEP stimulation paradigm, the chromatic stimulation results in a slightly worse accuracy, but with a lower danger for photosensitive epilepsy [AMR15].

As mentioned in a previous chapter, VEP BCIs are based on the idea of Sutter [Sut84], who stated that "the electrical scalp response to a modulated target is largest if the target is located within the central 1° of the visual field" and that "this makes it possible to construct a gaze-controlled keyboard". However, although the response to stimuli within the central 1° of the visual field is largest, Sutter has also shown that responses can be measured in the peripheral parts of the visual field [ST92]. In the case of a VEP BCI speller, this means that neighbored targets (letters) to the gazed target are also in the visual field and causes unwanted responses, which act as additional noise.

Considering the structure of the human eye, there two types of photoreceptor cells: cones and rods, which are differently distributed over the retina. Cones are mainly

located and densely packed in the central field and are responsible for color vision, whereas rods are mainly located at the outer parts of the retina and are used for peripheral vision [PAF⁺01]. Moreover, as shown by Bowmaker and Dartnall [BD80], cones are divided into three different types with different responses to different light wavelengths: the long-wave (L) cones, the middle-wave (M) cones, and the short-wave (S) cones with maximum responses at 564 nm, 534 nm, and 420 nm, respectively. Rods, on the other hand, have a maximum response at 498 nm.

Altogether, and the fact that there are approximately 91 million rod cells and only 4.5 million cone cells on the human retina [PAF⁺01], lead to the assumption that visual stimuli, excluding the wavelengths to which the rods respond, should lead to less peripheral perception and therefore to less noise.

To the best of my knowledge, yet it was not analyzed how strong the peripheral contribution is during a simultaneous stimulation of the central and peripheral visual field, which is the case with a BCI speller, for example. In this chapter, this issue will be addressed by analyzing the peripheral perception of different colors.

15.1 Methods

15.1.1 Analyze peripheral perception

To test the peripheral perception during a simultaneous stimulation of the central and peripheral visual field, a layout with nine equal-sized targets was used (Fig. 15.1). During the experiment, the center target is modulated with a 15 Hz SSVEP stimulation pattern. The eight outer targets are modulated with a 20 Hz SSVEP stimulation pattern. The participant has to focus solely on the center target.

Using the proposed layout, which could represent a section of the previously used matrix keyboard layout (Fig 4.1), in combination with the used SSVEP stimulation patterns, the peripheral perception can be analyzed using a frequency analysis. The lower the 20 Hz proportion, the lower is the peripheral perception. Contrary, the stronger the 15 Hz proportion, the stronger is the desired signal. The analysis was done with several stimulation colors, which will be explained in subsection 15.1.3.

15.1.2 Stimulation pattern

As mentioned, the center target is modulated with a 15 Hz SSVEP stimulation pattern, whereas the eight outer targets are modulated with a 20 Hz SSVEP stimulation pattern. For this, a binary sequence is generated which represents the corresponding stimulation pattern. The duration of an on- and off-stimulus is of equal length for the 15 Hz stimulation, whereas for the 20 Hz stimulation the on-stimulus is half of

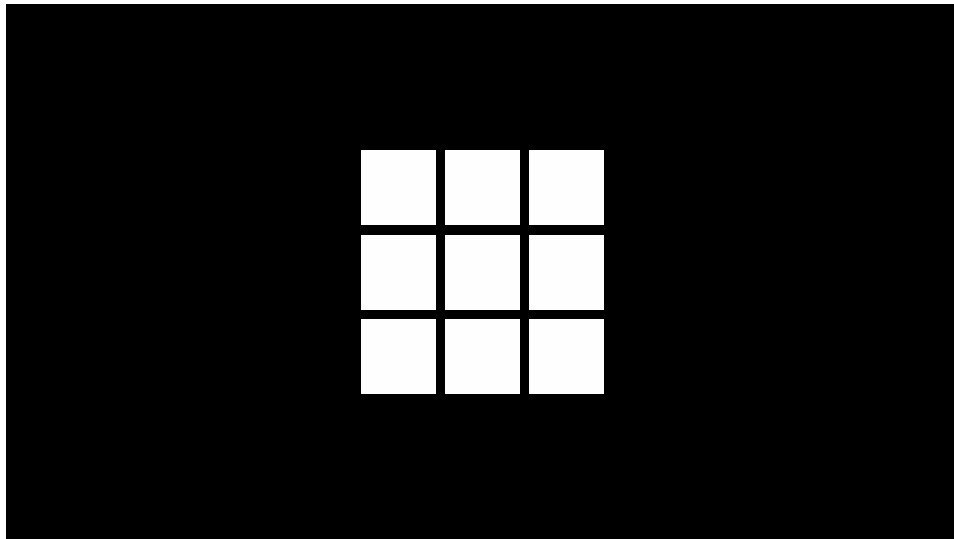


Figure 15.1: The layout used to identify peripheral perception. The center square is modulated with a 15 Hz SSVEP stimulation pattern, whereas the 8 outer squares are modulated with a 20 Hz SSVEP stimulation pattern.

the duration of the off-stimulus, which results in the following stimulation pattern for 1 s at the used presentation rate of 60 bit/s:

15 Hz: 11001100110011001100110011001100110011001100110011001100110011001100

20 Hz: 100







15.1.3 Color selection

As mentioned, the different cones on the retina respond differently to a different wavelength, which suggests using the wavelengths with the maximum response: 564 nm, 534 nm, and 420 nm. Furthermore, wavelengths around 498 nm should be avoided, to stimulate the rods as less as possible. On the other side, as many cones as possible should be stimulated to get the strongest response. This, and the fact that the absorption ranges of M- and L-cones overlap for most wavelengths, suggest using the wavelength where the M- and L-cones respond equally, which is at approximately 549 nm [BD80].

As a standard computer monitor is used for stimuli presentation, it cannot be guaranteed to emit only the desired wavelength. Therefore, an approximation is used by converting the wavelengths to RGB values using the tool by Dan Bruton [Bru96].

Also, red-green stimuli were tested, because an increased BCI performance was shown in previous studies. As a reference, white-black stimuli were tested, too. The complete set of the tested colors is shown in Table 15.1.

Table 15.1: Colors used for the analysis of the peripheral perception.

Color 1			Color 2		
Wavelength	RGB	Color	Wavelength	RGB	Color
–	(255,255,255)		–	(0,0,0)	
420 nm	(106,0,255)		–	(0,0,0)	
534 nm	(108,255,0)		–	(0,0,0)	
549 nm	(160,255,0)		–	(0,0,0)	
564 nm	(207,255,0)		–	(0,0,0)	
645 nm	(255,0,0)		510 nm	(0,255,0)	

The wavelength, RGB color code, as well as the corresponding color, are given. RGB values are approximated based on the wavelength using the tool by Dan Bruton [Bru96]. *Color 1* is displayed for a binary 1 of the stimulation pattern, whereas *Color 2* is displayed for a binary 0.

15.1.4 Analysis

To analyze the peripheral perception, a frequency analysis is performed. As the participant has to focus the center target which is modulated with a 15 Hz SSVEP pattern, this is the desired frequency, and the power should be as high as possible. Contrary, the eight outer targets are modulated with a 20 Hz SSVEP pattern, therefore, the higher the 20 Hz power is, the higher is the peripheral perception.

The power spectrum is calculated using the discrete Fourier transformation, which results in the magnitude M_k for each frequency bin $k \in 1 \dots N$ based on the signal X of length N :

$$M_k = \left| \sum_{j=1}^N (X_j \cdot e^{-j \cdot 2 \cdot \pi \cdot (k-1) \cdot (j-1) / n}) \right| \quad \forall k \in 1 \dots N \quad (15.1)$$

The magnitudes M are transformed into decibel (dB) using the following equation:

$$Y = 20 \cdot \log_{10}(M) \quad [\text{dB}] \quad (15.2)$$

As the trials have a length of 1 s (see section 15.1.6), each frequency bin k is equivalent to a frequency of k Hz, which ranges from 1 Hz to $N/2$ Hz.

The power alone is meaningless for the current analysis, instead, the power-gain ΔY compared to the baseline B is the desired value. The baseline B_k of the corresponding frequency k is approximated using the mean between the adjacent frequencies.

$$\Delta Y_k = Y_k - B_k \quad (15.3)$$

$$\text{where: } B_k = (Y_{k-1} + Y_{k+1}) / 2 \quad (15.4)$$

Furthermore, the ratio $R = \Delta Y_{15 \text{ Hz}} / \Delta Y_{20 \text{ Hz}}$ is calculated, as an increased ratio indicates an increased 15 Hz power and/or a decreased 20 Hz power, which is desired.

Also, the average results of each colored stimulus are compared to the results of the white-black stimulus, by calculating the corresponding difference. The higher the difference, the superior the colored stimulus is.

15.1.5 Hardware & Software

The setup is similar to the previous experiments (see section 11.1.3). The system consists of a g.USBamp (g.tec, Austria) EEG amplifier, two personal computers (PCs), Brainproducts Acticap system with 32 channels and an LCD monitor (BenQ XL2430-B) for stimuli presentation. Participants are seated approximately 80 cm in front of the monitor.

PC1 is used for the presentation on the LCD monitor, which is set to refresh rate of 60 Hz and its native resolution of 1920×1080 pixels. The layout (Fig. 15.1) was implemented using the framework described in chapter 4. A stimulus is represented by 0 or 1 in a binary sequence and is synchronized with the refresh rate of the LCD monitor. The timings of the monitor refresh cycles are synchronized with the EEG amplifier by using the parallel port.

PC2 is used for data acquisition, whereby BCI2000 [SMH⁺04] is used as a general framework for recording the data of the EEG amplifier. The amplifier sampling rate was set to 600 Hz, resulting in 10 samples per frame/stimulus. The data processing is done offline using MATLAB [MAT17].

A 32 electrodes EEG layout was used, 30 electrodes were located at Fz, T7, C3, Cz, C4, T8, CP3, CPz, CP4, P5, P3, P1, Pz, P2, P4, P6, PO9, PO7, PO3, POz, PO4, PO8, PO10, O1, POO1, POO2, O2, OI1h, OI2h, and Iz. The remaining two electrodes were used for electrooculography (EOG), one between the eyes and one left of the left eye. The ground electrode (GND) was positioned at FCz and reference electrode (REF) at OZ.

15.1.6 Data acquisition

Five participants were recruited to test the peripheral perception. The participants had to perform two runs for each color pair shown in Table. 15.1, whereby *Color 1* and *Color 2* are shown depending on the binary sequence of the stimulation pattern. Each run consists of 120 trials with a trial time of 1 s and without inter-trial time. Therefore, this results in a total of 240 trials for each color pair.

Prior to the experiment, a participant was introduced to focus only on the center target (Fig.15.1). A run was started by the participant with a delay of 5 s after pressing

the *space* key of the keyboard, therefore, the time between 2 runs was chosen by the participants themselves. To exclude temporal effects, the runs were performed in randomized order.

15.1.7 Preprocessing

The recorded EEG data is bandpass filtered by the amplifier between 0.1 Hz and 60 Hz using a Chebyshev filter of order 8 and an additional 50 Hz notch filter was applied.

To improve the signal-to-noise ratio, a spatial filter is applied to each trial using a canonical correlation analysis (CCA) similar to the method described in [SWRB14]. Only finding the best channel is different. The best channel is defined as the channel that correlates most with the 15 Hz stimulation pattern. To take the phase shift into account, the stimulation pattern is shifted sample-wise within the range of a full phase (2π). Afterward, the spatial filter is trained as described in [SWRB14] and applied to every single trial prior to the analysis.

15.2 Results

The frequency spectra are calculated for each subject and each color pair. Fig. 15.2 depicts the spectra for subject S1 including the approximated baselines and power-gains. The power-gains for the white, 534 nm, 549 nm, and 564 nm are quite similar. Contrary, using 420 nm results in a highly reduced 20 Hz power-gain. Interestingly, using the red-green color pair results in a reduced power-gain for both the 15 Hz and 20 Hz frequencies, respectively.

Detailed results for each subject are listed in Table 15.2, whereby the best values are highlighted in bold font. When using the white-black, as well as the 534 nm-black color pair, results in the highest average 15 Hz power-gain of 6.6 dB, which is desired. Contrary, the white-black color pair also results in the highest 20 Hz power-gain of 9.7 dB, which indicates a high peripheral perception. In fact, except for the 420 nm-black color pair, the average 20 Hz power-gain is even stronger than the average 15 Hz power-gain, which can easily be seen by the corresponding ratios. Interestingly, for S2 and S3 the 20 Hz power-gain is significantly stronger than the 15 Hz power-gain, for all color-pairs.

Conversely, the 420 nm-black color pair results in the best ratio of 1.28 between the 15 Hz power-gain and the 20 Hz power-gain. Especially S1 and S5 achieved a ratio of 4.28 and 3.06, respectively. Here it is worth to note, that the 15 Hz power-gain must also be considered, as a low power-gain corresponds to a weak VEP response. Unexpectedly, comparing the red-green color pair to the white-black reveals a significant decrease of power for both the 15 Hz and 20 Hz frequencies, respectively.

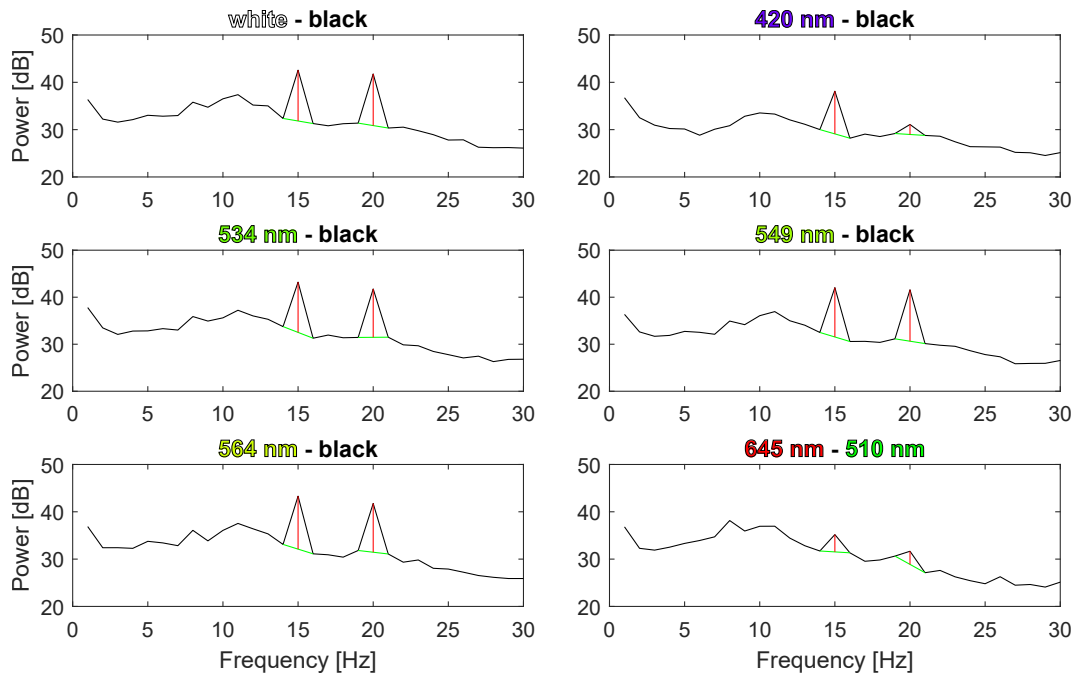


Figure 15.2: Frequency spectra of subject S1 for different colored stimuli. The black line represents the power in decibel (dB) for each frequency in Hertz (Hz). The green lines are the approximated base lines, whereas the red lines correspond to the power-gain. The 15 Hz stimulation pattern was focused by the subject, whereas the 20 Hz stimulation pattern was in the peripheral visual field.

15.3 Discussion

Most of recent VEP based BCIs make use of alternating white-black stimuli [CWN⁺15, BGW⁺11, SRB12a]. However, several color pairs were tested [DBR⁺14, WFL16, YXL⁺17, NSE15, AMR15], some superior to white-black and some not. Since VEP BCIs have generally several targets, which are close together, the adjacent targets to a desired target are also in the visual field and cause additional noise to the measured EEG. This peripheral perception was analyzed in the current chapter.

Since only five subjects were tested and the results are very different, the results must be interpreted with caution. Furthermore, as it cannot be guaranteed that the used monitor emits the desired wavelengths, the results should, therefore, serve as a basis for a further analysis.

As expected, in average the white-black color pair results in the strongest desired VEP response (15 Hz), but also in the strongest peripheral perception (20 Hz). As mentioned, the rods have a maximum absorption rate at a wavelength of 498 nm, therefore, lower/higher wavelengths should result in less peripheral perception. Compared to white, which is a mixture of all wavelengths, using 534 nm, 549 nm, and 564 nm results in a decreased power-gain of -0.2 dB, -0.6 dB, and -0.8 dB.

Table 15.2: Peripheral perception of different colors

Color pair	Power gain	Subject					mean	Difference to white
		S1	S2	S3	S4	S5		
white black	15 Hz [dB]	10.8	4.7	2.4	9.7	5.7	6.6	0
	20 Hz [dB]	10.9	13.5	10.3	6.9	6.8	9.7	0
	Ratio	0.98	0.35	0.23	1.41	0.84	0.68	0
420 nm black	15 Hz [dB]	9.0	2.0	4.4	4.5	3.6	4.7	-1.9
	20 Hz [dB]	2.1	5.7	6.1	3.4	1.2	3.7	-6.0
	Ratio	4.28	0.35	0.72	1.35	3.06	1.28	0.59
534 nm black	15 Hz [dB]	10.7	4.4	4.0	8.2	5.4	6.6	-0.1
	20 Hz [dB]	10.3	14.6	11.0	7.0	4.5	9.5	-0.2
	Ratio	1.04	0.30	0.37	1.18	1.22	0.69	0.01
549 nm black	15 Hz [dB]	10.5	4.7	2.7	9.4	5.4	6.5	-0.1
	20 Hz [dB]	10.9	12.3	9.5	6.7	5.9	9.1	-0.6
	Ratio	0.96	0.38	0.28	1.41	0.92	0.72	0.03
564 nm black	15 Hz [dB]	11.2	3.0	3.4	7.7	3.3	5.7	-0.9
	20 Hz [dB]	10.3	11.1	10.2	6.8	6.2	8.9	-0.8
	Ratio	1.09	0.27	0.33	1.14	0.54	0.64	-0.04
645 nm 510 nm	15 Hz [dB]	3.7	2.9	1.9	0.5	1.5	2.1	-4.6
	20 Hz [dB]	2.8	3.6	5.7	-0.4	1.8	2.7	-7.0
	Ratio	1.32	0.80	0.33	-1.45	0.85	0.78	0.09

Shown is the power-gain in dB of both the 15 Hz and 20 Hz frequencies. The power-gain is defined as the gain of power compared to the base line power, whereby the base line power is approximated by averaging the power of the adjacent frequencies. Furthermore, the ratio between the 15 Hz power-gain and the 20 Hz power-gain is given, as the ratio increases for less peripheral perception. Additionally, to compare each color pair to the white-black stimuli, the difference of the average power-gain as well as of the average ratio is given. The best values are highlighted in bold font, whereby "best" means the biggest power-gain for the 15 Hz frequency, the smallest power-gain for the 20 Hz frequency, and the biggest ratio between both, respectively.

Furthermore, using 420 nm results in an average decrease of -6.0 dB which corresponds to a loss of power-gain by 61.9% compared to white. However, it must be mentioned that also the 15 Hz power-gain is reduced by 28.8%, but in turn, this results in an average ratio of 1.28 compared to 0.68 for white. Here it is worth to note that S1 achieved a ratio of 4.28 with only -16.6% loss of the 15 Hz power-gain compared to white, this can easily be seen in Fig. 15.2, and implies that the peripheral perception is highly reduced. The only counterexample is subject S4, who achieved a better ratio using the white-black color pair.

Contrary to the findings of Yan *et al.* [YXL⁺17] and Nezamfar *et al.* [NSE15], the red-green color pair resulted in a significantly worse power-gain compared to white, for the 15 Hz as well as for the 20 Hz frequencies, which implies that the overall VEP

response is highly reduced. However, here it is worth to mention that the presented results are based on flash VEPs, whereas Yan *et al.* and Nezamfar *et al.* used a pattern-reversal checkerboard, which evokes different VEPs (see 2.3).

To conclude, as the 420 nm-black color pair results in the best ratio and the fact that all subjects mentioned that 420 nm causes less fatigue, suggest testing this color pair for a BCI application. Additionally, also the greenish wavelengths could be used, although the results are similar to the white-black color pair, the green color is more pleasant for the eyes.

Chapter 16

Summary

In this chapter, all the methods and results of this work are summarized to get a condensed overview before the final discussion. In advance, one of the main purposes of BCIs is to restore communication of motor disabled people, like ALS patients. As explained in section 3.3, a practical BCI should be most importantly effective (good accuracy), efficient (fast classifications), ease of use, and should allow performing as many different tasks as possible. Furthermore, it should also allow being used independently at home.

The goal of this work was to develop an BCI which address all those criteria, whereby the main focus was the independent usage. Prior to achieving this goal, in chapter 4 a stimulation framework was proposed which allows to easily create different stimulation layouts and ensures high synchronicity between the stimuli and the measured EEG. The effects of inaccurate synchronization to the performance of VEP BCIs was shown in chapter 5.

In chapter 6 a completely new and unique stimulation paradigm was proposed, based on fully random stimulation patterns. Based on this, a method (EEG2Code) was proposed which can predict arbitrary stimulation patterns based on the measured VEP responses. The resultant passive BCI showed that the visual information contained in the EEG could be predicted with an average ITR of 232 bit/min and up to 390 bit/min.

In chapter 7 the backward approach (Code2EEG) was proposed. It was shown that the variance in the brain response to arbitrary stimulation patterns could be explained by up to 25.9% on a single trial basis. It was further shown that this approach allows predicting the brain response to common stimulation paradigms, like SSVEP and cVEP, although the model was never trained on such stimuli.

In chapter 8 an optimized set of random stimulation patterns was proposed which increases the performance compared to fully random stimulation patterns.

In chapter 9 it was proven that the EEG2Code, as well as the Code2EEG approach,

can be used for synchronous BCI control, with accuracies of 97.9% and 93.9%, respectively. It was further shown that the optimized stimulation patterns also increase the active BCI performance, although the results were not significant due to the ceiling effect. Furthermore, a within-subject comparison has proven that the EEG2Code approach can compete with a state-of-the-art cVEP BCI approach.

In chapter 10 it was shown that a reduced trial duration improves the synchronous BCI performance to an average ITR of 154 bit/min and up to 231 bit/min. Furthermore, the main advantage of the proposed methods and the proposed stimulation paradigm were addressed. It was proven that both, the EEG2Code and Code2EEG approach, can be used with a nearly unlimited number of targets, which allows performing as many different tasks as required.

In chapter 11 the EEG2Code approach was extended to allow asynchronous control. In an online experiment, it was shown that the asynchronous approach achieves an average accuracy of 99.3% with classification durations of 2.1 seconds on average. Furthermore, it was also shown in an online experiment that the layout could be switched to more targets, for example, a user-friendly German QWERTY layout, which allowed to users to spell up to 29.7 correct case-sensitive letters per minute with an average accuracy of 98.1%. This proves that the approach is effective and efficient. To prove that the system can also be used independently, at least after the system was started, it was shown that the non-control state can be determined perfectly under four different conditions for 8 of 10 participants, whereas the intended control state was always detected for all participants.

In chapter 12 the mobile usage was addressed. As an exact synchronization can most probably not be achieved by using mobile devices (like wireless EEG electrodes), the performance of the EEG2Code model was analyzed in respect to varying synchronization latencies. It was shown that latencies in the range of up to 10 ms reduce the accuracy of the asynchronous approach by only 1%.

In chapter 13 the EEG2Code model was improved by using a convolution deep neural network. Although the training took slightly longer compared to the ridge regression model, the classification can still be performed in real-time. The results have shown that the asynchronous spelling performance can be increased to 176 bit/min with nearly the same accuracy.

In chapter 14 the performance using fewer training data was analyzed. The cross-subject analysis with no subject-dependent training data resulted in a drastically reduced performance. Contrary, reducing the training phase to 2 min, resulted in an accuracy loss of less than 1%. This would improve the ease of use criteria.

In chapter 15 it was proposed to further analyze the BCI performance in respect to the used stimuli color pair, as the results have shown that other color pairs than white-black reduce the peripheral perception of neighbored targets which, in turn, should improve the BCI performance.

Chapter 17

Discussion

As stated at the beginning of this thesis, being trapped in one's own body and especially no longer being able to communicate with relatives must be awful. Today's technology allows to restore the ability to communicate by interpreting brain signals. Although recent BCI spellers show high communication speed, they are not suitable for real-world applications. This was addressed in this work by developing a BCI which allows high-speed asynchronous control and which efficiently detects the user's intent to control the system or not.

The proposed asynchronous BCI approach has shown to be effective, efficient, ease of use, and allows to perform a virtually unlimited number of tasks. However, although it is a big step for moving BCIs out of the lab, there are further points that can and must be optimized in order to be suitable for end-user home-control. While the individual parts of this thesis were already discussed, the final discussion will focus on the issues that must be further addressed for end-user home-control as well as how the performance could be further increased.

The first and most important issue is that the system was tested with healthy participants. Although the presented BCI only requires a (corrected to) normal vision and the ability to voluntary control eye movements, it cannot be guaranteed that the system will achieve the same performance for end-user, like patients having amyotrophic lateral sclerosis (ALS). Unfortunately, there are virtually no studies comparing the performance difference of VEP based BCIs between healthy and impaired peoples. To the best of my knowledge, the only studies are by Lim *et al.* [LHH⁺13] and Hsu *et al.* [HLT⁺16], but no state-of-the-art methods were used, and only 1 and 3 ALS patients were analyzed, respectively, which is why no significant statement can be made. There are some studies, evaluating the VEP generation of ALS patients, with the result that some patients show slight abnormalities compared to healthy people [REE86, MTN⁺98]. However, this does not necessarily mean that the BCI performance will be reduced for such patients. What might be a problem, though, is that some studies have reported that about 45% of ALS patients reveal some form of

cognitive impairment [AFT⁺97, RSG⁺06]. Contrary, McCane *et al.* have found no significant performance difference between 14 ALS patients and 14 healthy people using a P300 speller [MHM⁺15].

Moreover, the performance was assessed under laboratory conditions. Although the same conditions could be achieved at home, it would be desirable to use a BCI on-the-go, which would most probably cause a decreased signal-to-noise ratio. For example, movements will add additional noise to EEG recordings, or solar radiation will reduce the contrast of the stimuli which in turn reduces VEP amplitudes.

Furthermore, the experiments were performed using gel electrodes, which requires time-consuming preparation and the gel dries out after a certain time. Contrary, dry electrodes can be used more easily and quickly, therefore, they are more convenient for end-users. As it was recently shown that dry electrodes are suitable for high-speed BCI control [Spü17, XWP⁺18], it would be interesting to assess the performance of the proposed BCI approach, especially using wireless electrodes, as it was shown that the approach could efficiently handle varying synchronization latencies.

Also, some possible optimizations have been mentioned. First and foremost, the non-control detection should be optimized to avoid absolutely all classifications during the non-control state, as random classifications decrease the user-experience and could have negative effects depending on the application purpose. For example, if the system is used to control lights, then it would surely not be desired that they switch themselves on and off again and again. As mentioned in chapter 11 this could be achieved by using a stricter threshold. While this would also reduce the spelling performance, it is suggested to use two thresholds: one optimized for the intentional control and one optimized for the non-control state.

Another optimization would be to make the BCI adaptive, because the EEG signals are non-stationary over time, even within the same user [GW11]. Adaptive BCI approaches have also shown to increase the performance [BRW⁺07, SRB12b, LGK14]. As mentioned in chapter 11 with the high average accuracy of the presented approach, it would be possible to implement a semi-supervised adaption of the EEG2Code model, whereby the model will be adaptively trained on test data which will be labeled with the predicted labels. Furthermore, the stimulation patterns could be adaptively optimized, whereby each stimulation pattern might be rated depending on the resultant trial length, the shorter a trial duration, the more likely the corresponding stimulation pattern will be used again.

To conclude, a general model for the prediction of arbitrary sensory stimuli was developed which allows discriminating 500,000 different visual stimuli based on 2 seconds of EEG data with an accuracy of up to 100 %. It was shown that the presented method could extract much more information from the EEG as can be used for BCI control. The observed ceiling effect allows the conclusion to be drawn that more powerful methods for brain signal decoding do not necessarily translate into a substantially better BCI control anymore. Furthermore, the presented asynchronous

BCI speller outperforms the previous state-of-the-art in several points. On the one hand, it increased both the spelling speed and the non-control state detection by a multi-fold. On the other hand, it is the most flexible system to-date, as it allows to use a virtually unlimited number of targets. By addressing the above-mentioned issues, the presented approach could pave the way to the very first end-user suitable BCI that can be used for any application purpose.

List of Abbreviations

ACC	Accuracy
ALS	Amyotrophic lateral sclerosis
BCI	Brain-Computer Interface
bpm	Bits per minute
CCA	Canonical correlation analysis
CNN	Convolutional neural network
CRT	Cathode ray tube
cVEP	Code-modulated visual-evoked potential
EEG	Electroencephalography
EOG	Electrooculography
ERP	Event-related potential
ErrP	Error-related potential
FFT	Fast fourier transformation
GND	Ground electrode
IC	Intentional control
ITR	Information transfer rate
JFPM	joint frequency-phase modulation
LCD	Liquid crystal display
LDA	Linear discriminant analysis
LED	Light emmiting diode
NC	Non control
OCSVM	One-class support vector machine
PC	Personal computer
REF	Reference electrode
ReLU	rectified linear unit
SSVEP	Steady-state visual-evoked potential
SVM	Support vector machine
SWDA	Step-wise discriminant analysis
SD	Standard deviation
VEP	Visual-evoked potential

Bibliography

- [ABG18] Eda Akman Aydin, Omer Faruk Bay, and Inan Guler. P300-based asynchronous brain computer interface for environmental control system. *IEEE journal of biomedical and health informatics*, 22(3):653–663, 2018.
- [AC16] John Atkinson and Daniel Campos. Improving bci-based emotion recognition by combining eeg feature selection and kernel classifiers. *Expert Systems with Applications*, 47:35–41, 2016.
- [ACG⁺09] Frederico AC Azevedo, Ludmila RB Carvalho, Lea T Grinberg, José Marcelo Farfel, Renata EL Ferretti, Renata EP Leite, Wilson Jacob Filho, Roberto Lent, and Suzana Herculano-Houzel. Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain. *Journal of Comparative Neurology*, 513(5):532–541, 2009.
- [AFT⁺97] Kazuo Abe, Harutoshi Fujimura, Keiko Toyooka, Saburo Sakoda, Shiro Yorifuji, and Takehiko Yanagihara. Cognitive function in amyotrophic lateral sclerosis. *Journal of the neurological sciences*, 148(1):95–100, 1997.
- [AGA⁺04] H Aurlien, IO Gjerde, JH Aarseth, G Eldøen, B Karlsen, H Skeidsvoll, and NE Gilhus. Eeg background activity described by a large computerized database. *Clinical Neurophysiology*, 115(3):665–673, 2004.
- [AMR15] Daiki Aminaka, Shoji Makino, and Tomasz M Rutkowski. Chromatic and high-frequency cvep-based bci paradigm. *arXiv preprint arXiv:1506.04461*, 2015.
- [ASA⁺11] F Aloise, F Schettini, P Aricò, F Leotta, S Salinari, D Mattia, F Babiloni, and F Cincotti. P300-based brain–computer interface for environmental control: an asynchronous approach. *Journal of neural engineering*, 8(2):025025, 2011.

- [BA03] Kenneth P Burnham and David R Anderson. *Model selection and multimodel inference: a practical information-theoretic approach*. Springer Science & Business Media, 2003.
- [BB02] Giedrius T Buračas and Geoffrey M Boynton. Efficient design of event-related fmri experiments using m-sequences. *Neuroimage*, 16(3):801–813, 2002.
- [BD80] James K Bowmaker and HJk Dartnall. Visual pigments of rods and cones in a human retina. *The Journal of physiology*, 298(1):501–511, 1980.
- [Ber29] Hans Berger. Über das elektrenkephalogramm des menschen. *Archiv für psychiatrie und nervenkrankheiten*, 87(1):527–570, 1929.
- [BFN11] David Beukelman, Susan Fager, and Amy Nordness. Communication support for people with als. *Neurology Research International*, 2011.
- [BGV92] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.
- [BGW⁺09] Guangyu Bin, Xiaorong Gao, Yijun Wang, Bo Hong, and Shangkai Gao. Vep-based brain-computer interfaces: time, frequency, and code modulations [research frontier]. *IEEE Computational Intelligence Magazine*, 4(4), 2009.
- [BGW⁺11] Guangyu Bin, Xiaorong Gao, Yijun Wang, Yun Li, Bo Hong, and Shangkai Gao. A high-speed bci based on code modulation vep. *Journal of neural engineering*, 8(2):025015, 2011.
- [BMK⁺06] Benjamin Blankertz, K-R Muller, Dean J Krusienski, Gerwin Schalk, Jonathan R Wolpaw, Alois Schlogl, Gert Pfurtscheller, Jd R Millan, Michael Schroder, and Niels Birbaumer. The bci competition iii: Validating alternative approaches to actual bci problems. *IEEE transactions on neural systems and rehabilitation engineering*, 14(2):153–159, 2006.
- [Bru96] Dan Bruton. Approximate rgb values for visible wavelengths. <http://www.physics.sfasu.edu/astro/color/spectra.html>, 1996.
- [BRW⁺07] Julie Blumberg, Jorn Rickert, Stephan Waldert, Andreas Schulze-Bonhage, Ad Aertsen, and Carsten Mehring. Adaptive classification for brain computer interfaces. In *Engineering in Medicine and*

- Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pages 2536–2539. IEEE, 2007.
- [BV97] David H Brainard and Spatial Vision. The psychophysics toolbox. *Spatial vision*, 10:433–436, 1997.
- [C+15] François Chollet et al. Keras. `keras.io`, 2015.
- [Cat75] Richard Caton. Electrical currents of the brain. *The Journal of Nervous and Mental Disease*, 2(4):610, 1875.
- [CC14] Hubert Cecotti and Damien Coyle. Calibration-less detection of steady-state visual evoked potentials-comparisons and combinations of methods. In *Neural Networks (IJCNN), 2014 International Joint Conference on*, pages 4050–4055. IEEE, 2014.
- [CCA+16] Jaiber Cardona, Eduardo Caicedo, Wilfredo Alfonso, Ricardo Chavarriaga, and José del R Millán. Superposition model for steady state visually evoked potentials. In *Systems, Man, and Cybernetics (SMC), 2016 IEEE International Conference on*, pages 004477–004482. IEEE, 2016.
- [Cec10] Hubert Cecotti. A self-paced and calibration-less ssvep-based brain-computer interface speller. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 18(2):127–133, April 2010.
- [CG11] Hubert Cecotti and Axel Graser. Convolutional neural networks for p300 detection with application to brain-computer interfaces. *IEEE transactions on pattern analysis and machine intelligence*, 33(3):433–445, 2011.
- [CGGX02] Ming Cheng, Xiaorong Gao, Shangkai Gao, and Dingfeng Xu. Design and implementation of a brain-computer interface with high transfer rates. *IEEE transactions on biomedical engineering*, 49(10):1181–1186, 2002.
- [CLN85] GE Chatrian, E Lettich, and PL Nelson. Ten percent electrode system for topographic studies of spontaneous and evoked eeg activities. *American Journal of EEG technology*, 25(2):83–92, 1985.
- [CM96] Gloria L Calhoun and Grant R McMillan. Eeg-based control for human-computer interaction. In *Human Interaction with Complex Systems, 1996. HICS'96. Proceedings., Third Annual Symposium on*, pages 4–9. IEEE, 1996.

- [CPAD⁺11] Almudena Capilla, Paula Pazo-Alvarez, Alvaro Darriba, Pablo Campo, and Joachim Gross. Steady-state visual evoked potentials can be explained by temporal superposition of transient event-related responses. *PLOS ONE*, 6(1):1–15, 01 2011.
- [CWD⁺13] Jennifer L Collinger, Brian Wodlinger, John E Downey, Wei Wang, Elizabeth C Tyler-Kabara, Douglas J Weber, Angus JC McMorland, Meel Velliste, Michael L Boninger, and Andrew B Schwartz. High-performance neuroprosthetic control by an individual with tetraplegia. *The Lancet*, 381(9866):557–564, 2013.
- [CWN⁺15] Xiaogang Chen, Yijun Wang, Masaki Nakanishi, Xiaorong Gao, Tzyy-Ping Jung, and Shangkai Gao. High-speed spelling with a noninvasive brain–computer interface. *Proceedings of the national academy of sciences*, 112(44):E6058–E6067, 2015.
- [CWNJ18] Kuan-Jung Chiang, Chun-Shu Wei, Masaki Nakanishi, and Tzyy-Ping Jung. Cross-subject transfer learning on high-speed steady-state visual evoked potential-based brain-computer interface. *arXiv preprint arXiv:1810.02842*, 2018.
- [CWW⁺11] Teng Cao, Xin Wang, Boyu Wang, Chi Man Wong, Feng Wan, Peng Un Mak, Pui In Mak, and Mang I Vai. 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, 2011.
- [DBR⁺14] Anna Duszyk, Maria Bierzyńska, Zofia Radzikowska, Piotr Milanowski, Rafał Kuś, Piotr Suffczyński, Magdalena Michalska, Maciej Łabęcki, Piotr Zwoliński, and Piotr Durka. Towards an optimization of stimulus parameters for brain-computer interfaces based on steady state visual evoked potentials. *Plos one*, 9(11):e112099, 2014.
- [DKF06] Olivier David, James M Kilner, and Karl J Friston. Mechanisms of evoked and induced responses in meg/eeg. *Neuroimage*, 31(4):1580–1591, 2006.
- [DMPL11] Pablo F Diez, Vicente A Mut, Enrique M Avila Perona, and Eric Lacia Leber. Asynchronous bci control using high-frequency ssvep. *Journal of neuroengineering and rehabilitation*, 8(1):39, 2011.
- [DSMM10] Bernardo Dal Seno, Matteo Matteucci, and Luca T Mainardi. The utility metric: a novel method to assess the overall performance of discrete brain–computer interfaces. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 18(1):20–28, 2010.

- [Elz10] Tobias Elze. Achieving precise display timing in visual neuroscience experiments. *Journal of neuroscience methods*, 191(2):171–179, 2010.
- [FD88] Lawrence Ashley Farwell and Emanuel Donchin. Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalography and clinical Neurophysiology*, 70(6):510–523, 1988.
- [Fre70] Paulo Freire. Pedagogy of the oppressed (MB Ramos, trans.). *The Continuum International Publishing Group*, 2007:77, 1970.
- [FVG07] Ola Friman, Ivan Volosyak, and Axel Graser. Multiple channel detection of steady-state visual evoked potentials for brain-computer interfaces. *IEEE transactions on biomedical engineering*, 54(4):742–750, 2007.
- [GDS⁺09] Christoph Guger, Shahab Daban, Eric Sellers, Clemens Holzner, Gunther Krausz, Roberta Carabalona, Furio Gramatica, and Guenter Edlinger. How many people are able to control a p300-based brain-computer interface (bci)? *Neuroscience letters*, 462(1):94–98, 2009.
- [GNL⁺08] Ferran Galán, Marnix Nuttin, Eileen Lew, Pierre W Ferrez, Gerolf Vanacker, Johan Philips, and J del R Millán. A brain-actuated wheelchair: asynchronous and non-invasive brain-computer interfaces for continuous control of robots. *Clinical neurophysiology*, 119(9):2159–2169, 2008.
- [GP02] Craig J Gonsalvez and John Polich. P300 amplitude is determined by target-to-target interval. *Psychophysiology*, 39(3):388–396, 2002.
- [GSF⁺17] Sebastian Grissmann, Martin Spüler, Josef Faller, Tanja Krumpel, Thorsten Zander, Augustin Kelava, Christian Scharinger, and Peter Gerjets. Context sensitivity of eeg-based workload classification under different affective valence. *IEEE Transactions on Affective Computing*, 2017.
- [Gun18] Felix van Gunsteren. Deep neural networks for classification of eeg data. Master’s thesis, University of Tübingen, WSI, Tübingen, July 2018.
- [GVLdIM14] Pablo Garaizar, Miguel A Vadillo, Diego López-de Ipiña, and Helena Matute. Measuring software timing errors in the presentation of visual stimuli in cognitive neuroscience experiments. *PloS one*, 9(1):e85108, 2014.

- [GW11] Moritz Grosse-Wentrup. What are the causes of performance variation in brain-computer interfacing? *International Journal of Bioelectromagnetism*, 13, 01 2011.
- [HAAN13] Neda Haghighatpanah, Rasoul Amirfattahi, Vahid Abootalebi, and Behzad Nazari. A single channel-single trial p300 detection algorithm. In *Electrical Engineering (ICEE), 2013 21st Iranian Conference on*, pages 1–5. IEEE, 2013.
- [HBJ⁺12] Leigh R Hochberg, Daniel Bacher, Beata Jarosiewicz, Nicolas Y Masse, John D Simeral, Joern Vogel, Sami Haddadin, Jie Liu, Sydney S Cash, Patrick van der Smagt, et al. Reach and grasp by people with tetraplegia using a neurally controlled robotic arm. *Nature*, 485(7398):372, 2012.
- [HBKK15] Elisa Mira Holz, Loic Botrel, Tobias Kaufmann, and Andrea Kübler. Long-term independent brain-computer interface home use improves quality of life of a patient in the locked-in state: a case study. *Archives of physical medicine and rehabilitation*, 96(3):S16–S26, 2015.
- [Her01] Christoph S Herrmann. Human eeg responses to 1–100 hz flicker: resonance phenomena in visual cortex and their potential correlation to cognitive phenomena. *Experimental brain research*, 137(3-4):346–353, 2001.
- [HK70] Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.
- [HLJ⁺12] Han-Jeong Hwang, Jeong-Hwan Lim, Young-Jin Jung, Han Choi, Sang Woo Lee, and Chang-Hwan Im. Development of an ssvep-based bci spelling system adopting a qwerty-style led keyboard. *Journal of neuroscience methods*, 208(1):59–65, 2012.
- [HLT⁺16] Hao-Teng Hsu, I-Hui Lee, Han-Ting Tsai, Hsiang-Chih Chang, Kuo-Kai Shyu, Chuan-Chih Hsu, Hsiao-Huang Chang, Ting-Kuang Yeh, Chun-Yen Chang, and Po-Lei Lee. Evaluate the feasibility of using frontal ssvep to implement an ssvep-based bci in young, elderly and als groups. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 24(5):603–615, 2016.
- [HM02] J Hanagata and K Momose. A method for detecting gazed target using visual evoked potentials elicited by pseudorandom stimuli. In *Proc. 5th Asia Pacific Conf. Medical and Biological Engineering and 11th Int. Conf. Biomedical Engineering (ICBME)*, 2002.

- [HSF⁺06] Leigh R Hochberg, Mijail D Serruya, Gerhard M Friehs, Jon A Mukand, Maryam Saleh, Abraham H Caplan, Almut Branner, David Chen, Richard D Penn, and John P Donoghue. Neuronal ensemble control of prosthetic devices by a human with tetraplegia. *Nature*, 442(7099):164, 2006.
- [HSM09] SA Huettel, AW Song, and G McCarthy. Functional magnetic resonance imaging , massachusetts: Sinauer. Technical report, ISBN 978-0-87893-286-3, 2009.
- [HW62] David H Hubel and Torsten N Wiesel. Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex. *The Journal of physiology*, 160(1):106–154, 1962.
- [ICM⁺14] Iñaki Iturrate, Ricardo Chavarriaga, Luis Montesano, Javier Minguez, and JdR Millán. Latency correction of event-related potentials between different experimental protocols. *Journal of neural engineering*, 11(3):036005, 2014.
- [IS15] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.
- [JAK⁺04] Ben H Jansen, Anand Allam, Prashant Kota, Kathleen Lachance, Ayokunle Osho, and Karthik Sundaresan. An exploratory study of factors affecting single trial p300 detection. *IEEE transactions on Biomedical Engineering*, 51(6):975–978, 2004.
- [Jas58] Herbert H Jasper. The ten-twenty electrode system of the international federation. *Electroencephalogr. Clin. Neurophysiol.*, 10:370–375, 1958.
- [JDZ⁺14] Jing Jin, Ian Daly, Yu Zhang, Xingyu Wang, and Andrzej Cichocki. An optimized erp brain–computer interface based on facial expression changes. *Journal of neural engineering*, 11(3):036004, 2014.
- [KFH⁺09] Andrea Kübler, Adrian Furdea, Sebastian Halder, Eva Maria Hammer, Femke Nijboer, and Boris Kotchoubey. A brain–computer interface controlled auditory event-related potential (p300) spelling system for locked-in patients. *Annals of the New York Academy of Sciences*, 1157(1):90–100, 2009.
- [KK14] Tobias Kaufmann and Andrea Kübler. Beyond maximum speed—a novel two-stimulus paradigm for brain–computer interfaces based on event-related potentials (p300-bci). *Journal of neural engineering*, 11(5):056004, 2014.

- [KML17] No-Sang Kwak, Klaus-Robert Müller, and Seong-Whan Lee. A convolutional neural network for steady state visual evoked potential classification under ambulatory environment. *PloS one*, 12(2):e0172578, 2017.
- [KMP87] JJ Kulikowski, IJ Murray, and NR Parry. Human visual evoked potentials to chromatic and achromatic gratings. *Clin. Vis. Sci*, 1:231–244, 1987.
- [KRM96] JANUS J KULIKOWSKI, ANTHONY G ROBSON, and DECLAN J McKEEFY. Specificity and selectivity of chromatic visual evoked potentials. *Vision Research*, 36(21):3397–3401, 1996.
- [KSR⁺18] Tanja Krumpe, Christian Scharinger, Wolfgang Rosenstiel, Peter Gerjets, and Martin Spüler. Unity and diversity in working memory load: Evidence for the separability of the executive functions updating and inhibition using machine learning. *Biological psychology*, 139:163–172, 2018.
- [LBH15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.
- [LFMP⁺07] Robert Leeb, Doron Friedman, Gernot R Müller-Putz, Reinhold Scherer, Mel Slater, and Gert Pfurtscheller. Self-paced (asynchronous) bci control of a wheelchair in virtual environments: a case study with a tetraplegic. *Computational intelligence and neuroscience*, 2007, 2007.
- [LGK14] A Llera, Vicenç Gómez, and Hilbert J Kappen. Adaptive multi-class classification for brain computer interfaces. *Neural computation*, 26(6):1108–1127, 2014.
- [LGZ08a] Shijian Lu, Cuntai Guan, and Haihong Zhang. Subject-independent brain computer interface through boosting. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE, 2008.
- [LGZ08b] Shijian Lu, Cuntai Guan, and Haihong Zhang. Unsupervised brain computer interface based on inter-subject information. In *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, pages 638–641. IEEE, 2008.
- [LHH⁺13] Jeong-Hwan Lim, Han-Jeong Hwang, Chang-Hee Han, Ki-Young Jung, and Chang-Hwan Im. Classification of binary intentions for individuals with impaired oculomotor function: ‘eyes-closed’ ssvp-based brain–computer interface (bci). *Journal of neural engineering*, 10(2):026021, 2013.

- [LPR⁺06] Edmund C Lalor, Barak A Pearlmutter, Richard B Reilly, Gary Mc-Darby, and John J Foxe. The vespa: a method for the rapid estimation of a visual evoked potential. *Neuroimage*, 32(4):1549–1561, 2006.
- [LPWY13] Yuanqing Li, Jiahui Pan, Fei Wang, and Zhuliang Yu. A hybrid bci system combining p300 and ssvep and its application to wheelchair control. *IEEE Transactions on Biomedical Engineering*, 60(11):3156–3166, 2013.
- [LRL⁺14] Zulay R Lugo, Javi Rodriguez, Alexander Lechner, Rupert Ortner, Ithabi S Gantner, Steven Laureys, Quentin Noirhomme, and Christoph Guger. A vibrotactile p300-based brain–computer interface for consciousness detection and communication. *Clinical EEG and neuroscience*, 45(1):14–21, 2014.
- [LSW⁺18] Vernon Lawhern, Amelia Solon, Nicholas Waytowich, Stephen M Gordon, Chou Hung, and Brent J Lance. Eegnet: a compact convolutional neural network for eeg-based brain–computer interfaces. *Journal of neural engineering*, 2018.
- [LWL18] Yonghui Liu, Qingguo Wei, and Zongwu Lu. A multi-target brain-computer interface based on code modulated visual evoked potentials. *PloS one*, 13(8):e0202478, 2018.
- [MAT17] MATLAB. *version 9.3 (R2017b)*. The MathWorks Inc., Natick, Massachusetts, 2017.
- [MCVH12] Nikolay V Manyakov, Nikolay Chumerin, and Marc M Van Hulle. Multichannel decoding for phase-coded ssvep brain–computer interface. *International journal of neural systems*, 22(05):1250022, 2012.
- [MHH17] Mostafa Mohammadpour, Seyyed Mohammad Reza Hashemi, and Negin Houshmand. Classification of eeg-based emotion for bci applications. In *Artificial Intelligence and Robotics (IRANOPEN), 2017*, pages 127–131. IEEE, 2017.
- [MHM⁺15] Lynn M McCane, Susan M Heckman, Dennis J McFarland, George Townsend, Joseph N Mak, Eric W Sellers, Debra Zeitlin, Laura M Tenteromano, Jonathan R Wolpaw, and Theresa M Vaughan. P300-based brain-computer interface (bci) event-related potentials (erps): People with amyotrophic lateral sclerosis (als) vs. age-matched controls. *Clinical Neurophysiology*, 126(11):2124–2131, 2015.
- [MHT17] Md Rakibul Mowla, Jane E Huggins, and David E Thompson. Enhancing p300-bci performance using latency estimation. *Brain-Computer Interfaces*, 4(3):137–145, 2017.

- [MMV⁺12] Joseph N Mak, Dennis J McFarland, Theresa M Vaughan, Lynn M McCane, Phillippa Z Tsui, Debra J Zeitlin, Eric W Sellers, and Jonathan R Wolpaw. Eeg correlates of p300-based brain–computer interface (bci) performance in people with amyotrophic lateral sclerosis. *Journal of neural engineering*, 9(2):026014, 2012.
- [MN98] Makoto Matsumoto and Takuji Nishimura. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 8(1):3–30, 1998.
- [MNK⁺17] Lenis Meriño, Tapsya Nayak, Prasanna Kolar, Garrett Hall, Zijing Mao, Daniel J Pack, and Yufei Huang. Asynchronous control of unmanned aerial vehicles using a steady-state visual evoked potential-based brain computer interface. *Brain-Computer Interfaces*, 4(1-2):122–135, 2017.
- [Mom07] Keiko Momose. Evaluation of an eye gaze point detection method using vep elicited by multi-pseudorandom stimulation for brain computer interface. In *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pages 5063–5066. IEEE, 2007.
- [Moo03] Melody M Moore. Real-world applications for brain-computer interface technology. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 11(2):162–165, 2003.
- [MPSNP06] Gernot R Muller-Putz, Reinhold Scherer, Christa Neuper, and Gert Pfurtscheller. Steady-state somatosensory evoked potentials: suitable brain signals for brain-computer interfaces? *IEEE transactions on neural systems and rehabilitation engineering*, 14(1):30–37, 2006.
- [MRMK96] DJ McKeefry, MHA Russell, IJ Murray, and JJ Kulikowski. Amplitude and phase variations of harmonic components in human achromatic and chromatic visual evoked potentials. *Visual Neuroscience*, 13(4):639–653, 1996.
- [MSM⁺14] Lynn M McCane, Eric W Sellers, Dennis J McFarland, Joseph N Mak, C Steve Carmack, Debra Zeitlin, Jonathan R Wolpaw, and Theresa M Vaughan. Brain-computer interface (bci) evaluation in people with amyotrophic lateral sclerosis. *Amyotrophic lateral sclerosis and frontotemporal degeneration*, 15(3-4):207–215, 2014.
- [MTN⁺98] TF Münte, Mathias C Tröger, Isabel Nusser, Bernardina M Wieringa, Sönke Johannes, Mike Matzke, and Reinhard Dengler. Alteration

- of early components of the visual evoked potential in amyotrophic lateral sclerosis. *Journal of neurology*, 245(4):206–210, 1998.
- [MVD⁺15] Felip Miralles, Eloisa Vargiu, Stefan Dauwalder, Marc Solà, Ger- not Müller-Putz, Selina C Wriessnegger, Andreas Pinegger, Andrea Kübler, Sebastian Halder, Ivo Käthner, et al. Brain computer inter- face on track to home. *The Scientific World Journal*, 2015, 2015.
- [Nag19] Sebastian Nagel. Simplified application for visual experiments. doc- umentation. [https://github.com/nagel86/save-framework/ blob/master/documentation.pdf](https://github.com/nagel86/save-framework/blob/master/documentation.pdf), 2019.
- [NDRS18] Sebastian Nagel, Werner Dreher, Wolfgang Rosenstiel, and Martin Spüler. The effect of monitor raster latency on veps, erps and brain– computer interface performance. *Journal of neuroscience methods*, 295:45–50, 2018.
- [NRS17] Sebastian Nagel, Wolfgang Rosenstiel, and Martin Spüler. Random visual evoked potentials (rvep) for brain-computer interface (bci) control. In *Proceedings of the 7th International Brain-Computer In- terface Conference*, pages 349–354, 09 2017.
- [NRS18] Sebastian Nagel, Wolfgang Rosenstiel, and Martin Spüler. Finding optimal stimulation patterns for bcis based on visual evoked poten- tials. In *Proceedings of the 7th International BCI Meeting*, pages 164–165. BCI Society, 05 2018.
- [NS18] Sebastian Nagel and Martin Spüler. Modelling the brain response to arbitrary visual stimulation patterns for a flexible high-speed brain- computer interface. *PloS one*, 13(10):e0206107, 2018.
- [NS19a] Sebastian Nagel and Martin Spüler. Asynchronous non-invasive high-speed bci speller with robust non-control state detection. *Scien- tific reports*, 9(1):8269, 2019.
- [NS19b] Sebastian Nagel and Martin Spüler. World’s fastest brain-computer interface: Combining eeg2code with deep learning. *PLOS ONE*, 14(9):1–15, 09 2019.
- [NSE15] Hooman Nezamfar, Seyed Sadegh Mohseni Salehi, and Deniz Erdog- mus. Stimuli with opponent colors and higher bit rate enable higher accuracy for c-vep bci. In *Signal Processing in Medicine and Biology Symposium (SPMB), 2015 IEEE*, pages 1–6. IEEE, 2015.

- [OBB⁺16] J Vernon Odom, Michael Bach, Mitchell Brigell, Graham E Holder, Daphne L McCulloch, Atsushi Mizota, Alma Patrizia Tormene, International Society for Clinical Electrophysiology of Vision, et al. Iscev standard for clinical visual evoked potentials:(2016 update). *Documenta Ophthalmologica*, 133(1):1–9, 2016.
- [PAB⁺10] Gert Pfurtscheller, Brendan Z Allison, Günther Bauernfeind, Clemens Brunner, Teodoro Solis Escalante, Reinhold Scherer, Thorsten O Zander, Gernot Mueller-Putz, Christa Neuper, and Niels Birbaumer. The hybrid bci. *Frontiers in neuroscience*, 4:3, 2010.
- [PAF⁺01] D Purves, GJ Augustine, D Fitzpatrick, LC Katz, AS LaMantia, JO McNamara, and SM Williams. Anatomical distribution of rods and cones. *Neuroscience. Sunderland (MA): Sinauer Associates*, 2001.
- [Pea95] Karl Pearson. Note on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*, 58:240–242, 1895.
- [PFH⁺15] Andreas Pinegger, Josef Faller, Sebastian Halder, Selina Wriessneger, and Gernot Müller-Putz. Control or non-control state: That is the question! an asynchronous visual p300- based bci approach. *Journal of neural engineering*, 12:014001, 01 2015.
- [PGD⁺11] Jie Pan, Xiaorong Gao, Fang Duan, Zheng Yan, and Shangkai Gao. Enhancing the classification accuracy of steady-state visual evoked potential-based brain–computer interfaces using phase constrained canonical correlation analysis. *Journal of neural engineering*, 8(3):036027, 2011.
- [PMTA09] Sergio Parini, Luca Maggi, Anna C Turconi, and Giuseppe Andreoni. A robust and self-paced bci system based on a four class ssvp paradigm: algorithms and protocols for a high-transfer-rate direct brain communication. *Computational Intelligence and Neuroscience*, 2009, 2009.
- [Pol07] John Polich. Updating p300: an integrative theory of p3a and p3b. *Clinical neurophysiology*, 118(10):2128–2148, 2007.
- [PPS11] Rajesh C. Panicker, Sadasivan Puthusserypady, and Ying Sun. An asynchronous p300 bci with ssvp-based control state detection. *IEEE Transactions on Biomedical Engineering*, 58(6):1781–1788, June 2011.

- [PSD09] Ilias Pappas, Stylianos Siskos, and Charalambos A. Dimitriadis. Active-matrix liquid crystal displays - operation, electronics and analog circuits design. In Georgiy V Tkachenko, editor, *New Developments in Liquid Crystals*, chapter 08, pages 147–170. InTech, Rijeka, 2009.
- [Pyt10] Python. *version 2.7*. Python Software Foundation, Wilmington, Delaware, 2010.
- [REE86] Rodney A Radtke, Andrea Erwin, and CW Erwin. Abnormal sensory evoked potentials in amyotrophic lateral sclerosis. *Neurology*, 36(6):796–796, 1986.
- [Reg66] David Regan. An effect of stimulus colour on average steady-state potentials evoked in man. *Nature*, 210(5040):1056, 1966.
- [RFPN11] Boris Reuderink, Jason Farquhar, Mannes Poel, and Anton Nijholt. A subject-independent brain-computer interface based on smoothed, second-order baselining. In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 4600–4604. IEEE, 2011.
- [RP99] Daran Ravden and John Polich. On p300 measurement stability: habituation, intra-trial block variation, and ultradian rhythms. *Biological psychology*, 51(1):59–76, 1999.
- [RSC⁺94] Jeff Rabin, Eugene Switkes, Michael Crognale, Marilyn E Schneck, and Anthony J Adams. Visual evoked potentials in three-dimensional color space: correlates of spatio-chromatic processing. *Vision research*, 34(20):2657–2671, 1994.
- [RSG⁺06] Gregory A Rippon, Nikolaos Scarmeas, Paul H Gordon, Peregrine L Murphy, Steven M Albert, Hiroshi Mitsumoto, Karen Marder, Lewis P Rowland, and Yaakov Stern. An observational study of cognitive impairment in amyotrophic lateral sclerosis. *Archives of Neurology*, 63(3):345–352, 2006.
- [SBK⁺12] Martin Spüler, Michael Bensch, Sonja Kleih, Wolfgang Rosenstiel, Martin Bogdan, and Andrea Kübler. Online use of error-related potentials in healthy users and people with severe motor impairment increases performance of a p300-bci. *Clinical Neurophysiology*, 123(7):1328–1337, 2012.
- [Sch00] Albrecht Schmidt. Implicit human computer interaction through context. *Personal technologies*, 4(2-3):191–199, 2000.

- [Sha48] Claude Elwood Shannon. A mathematical theory of communication. *Bell system technical journal*, 27(3):379–423, 1948.
- [SK18] Martin Spüler and Simone Kurek. Alpha-band lateralization during auditory selective attention for brain–computer interface control. *Brain-Computer Interfaces*, 5(1):23–29, 2018.
- [SLS18] Hongchang Shan, Yu Liu, and Todor Stefanov. A simple convolutional neural network for accurate p300 detection and character spelling in brain computer interface. In *IJCAI*, pages 1604–1610, 2018.
- [SMH⁺04] Gerwin Schalk, Dennis J McFarland, Thilo Hinterberger, Niels Birbaumer, and Jonathan R Wolpaw. Bci2000: a general-purpose brain-computer interface (bci) system. *IEEE Transactions on biomedical engineering*, 51(6):1034–1043, 2004.
- [Spü17] Martin Spüler. A high-speed brain-computer interface (bci) using dry eeg electrodes. *PloS one*, 12(2):e0172400, 2017.
- [SRB12a] Martin Spüler, Wolfgang Rosenstiel, and Martin Bogdan. One class svm and canonical correlation analysis increase performance in a c-vep based brain-computer interface (bci). In *Proceedings of 20th European Symposium on Artificial Neural Networks (ESANN 2012)*, pages 103–108, Bruges, Belgium, 04 2012.
- [SRB12b] Martin Spüler, Wolfgang Rosenstiel, and Martin Bogdan. On-line adaptation of a c-vep brain-computer interface (bci) based on error-related potentials and unsupervised learning. *PloS one*, 7(12):e51077, 2012.
- [ST92] Erich E Sutter and Duong Tran. The field topography of erg components in man—i. the photopic luminance response. *Vision research*, 32(3):433–446, 1992.
- [ST18] Kaori Suefusa and Toshihisa Tanaka. Asynchronous brain–computer interfacing based on mixed-coded visual stimuli. *IEEE Transactions on Biomedical Engineering*, 65(9):2119–2129, 2018.
- [STK18] F Sobreira, C Tremmel, and DJ Krusienski. Modeling the visual pathway for stimulus optimization in brain-computer interfaces. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 1672–1675. IEEE, 2018.
- [Sut84] Erich E Sutter. The visual evoked response as a communication channel. In *Proceedings of the IEEE Symposium on Biosensors*, pages 95–100, 1984.

- [Sut92] Erich E Sutter. The brain response interface: communication through visually-induced electrical brain responses. *Journal of Microcomputer Applications*, 15(1):31–45, 1992.
- [SVW10] Eric W Sellers, Theresa M Vaughan, and Jonathan R Wolpaw. A brain-computer interface for long-term independent home use. *Amyotrophic lateral sclerosis*, 11(5):449–455, 2010.
- [SW03] George AF Seber and Christopher J Wild. *Nonlinear Regression*. John Wiley & Sons, Inc., Hoboken, New Jersey, September 2003.
- [SWRB14] Martin Spüler, Armin Walter, Wolfgang Rosenstiel, and Martin Bogdan. Spatial filtering based on canonical correlation analysis for classification of evoked or event-related potentials in eeg data. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 22(6):1097–1103, 2014.
- [SWST81] C Saunders, T Walsh, M Smith, and J Teller. Hospice care in the motor neuron diseases. *Hospice: The living idea*. London, UK: Edward Arnold, 1981.
- [SZ14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [SZG⁺07] Ranganatha Sitaram, Haihong Zhang, Cuntai Guan, Manoj Thulasidas, Yoko Hoshi, Akihiro Ishikawa, Koji Shimizu, and Niels Birbaumer. Temporal classification of multichannel near-infrared spectroscopy signals of motor imagery for developing a brain-computer interface. *NeuroImage*, 34(4):1416–1427, 2007.
- [TGW06] Manoj Thulasidas, Cuntai Guan, and Jiankang Wu. Robust classification of eeg signal for brain-computer interface. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 14(1):24–29, 2006.
- [TMS⁺17] John Thomas, Tomasz Maszczyk, Nishant Sinha, Tilmann Kluge, and Justin Dauwels. Deep learning-based classification for brain-computer interfaces. In *Systems, Man, and Cybernetics (SMC), 2017 IEEE International Conference on*, pages 234–239. IEEE, 2017.
- [TvdBFD15] Jordy Thielen, Philip van den Broek, Jason Farquhar, and Peter De-sain. Broad-band visually evoked potentials: re (con) volution in brain-computer interfacing. *PloS one*, 10(7):e0133797, 2015.
- [TZ15] Jijun Tong and Danhua Zhu. Multi-phase cycle coding for ssvep based brain-computer interfaces. *Biomedical engineering online*, 14(1):5, 2015.

- [Vid73] Jacques J Vidal. Toward direct brain-computer communication. *Annual review of Biophysics and Bioengineering*, 2(1):157–180, 1973.
- [Vid77] Jacques J Vidal. Real-time detection of brain events in eeg. *Proceedings of the IEEE*, 65(5):633–641, 1977.
- [VMS⁺06] Theresa M Vaughan, Dennis J McFarland, Gerwin Schalk, William A Sarnacki, Dean J Krusienski, Eric W Sellers, and Jonathan R Wolpaw. The wadsworth bci research and development program: at home with bci. *IEEE transactions on neural systems and rehabilitation engineering*, 14(2):229–233, 2006.
- [Vol11] Ivan Volosyak. Ssvep-based bremen–bci interface—boosting information transfer rates. *Journal of neural engineering*, 8(3):036020, 2011.
- [VPH⁺93] Arno Villringer, J Planck, C Hock, L Schleinkofer, and U Dirnagl. Near infrared spectroscopy (nirs): a new tool to study hemodynamic changes during activation of brain function in human adults. *Neuroscience letters*, 154(1-2):101–104, 1993.
- [WFG⁺16] Nicholas R Waytowich, Josef Faller, Javier O Garcia, Jean M Vettel, and Paul Sajda. Unsupervised adaptive transfer learning for steady-state visual evoked potential brain-computer interfaces. In *Systems, Man, and Cybernetics (SMC), 2016 IEEE International Conference on*, pages 004135–004140. IEEE, 2016.
- [WFL16] Qingguo Wei, Siwei Feng, and Zongwu Lu. Stimulus specificity of brain-computer interfaces based on code modulation visual evoked potentials. *PloS one*, 11(5):e0156416, 2016.
- [WGG15] Yijun Wang, Xiaorong Gao, and Shangkai Gao. Computational modeling and application of steady-state visual evoked potentials in brain-computer interfaces. *Sci Suppl*, 350(6256):43–46, 2015.
- [WLG⁺18] Qingguo Wei, Yonghui Liu, Xiaorong Gao, Yijun Wang, Chen Yang, Zongwu Lu, and Huayuan Gong. A novel c-vep bci paradigm for increasing the number of stimulus targets based on grouping modulation with different codes. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2018.
- [WMB⁺04] Nikolaus Weiskopf, Klaus Mathiak, Simon W Bock, Frank Scharnowski, Ralf Veit, Wolfgang Grodd, Rainer Goebel, and Niels Birbaumer. Principles of a brain-computer interface (bci) based on real-time functional magnetic resonance imaging (fmri). *IEEE transactions on biomedical engineering*, 51(6):966–970, 2004.

- [WMSW10] J Adam Wilson, Jürgen Mellinger, Gerwin Schalk, and Justin Williams. A procedure for measuring latencies in brain–computer interfaces. *IEEE transactions on biomedical engineering*, 57(7):1785–1797, 2010.
- [WNS⁺14] Armin Walter, Georgios Naros, Martin Spüler, Alireza Gharabaghi, Wolfgang Rosenstiel, and Martin Bogdan. Decoding stimulation intensity from evoked ecog activity. *Neurocomputing*, 141:46–53, 2014.
- [WRB⁺17] Carina Walter, Wolfgang Rosenstiel, Martin Bogdan, Peter Gerjets, and Martin Spüler. Online eeg-based workload adaptation of an arithmetic learning environment. *Frontiers in human neuroscience*, 11:286, 2017.
- [WRMP98] Jonathan R Wolpaw, Herbert Ramoser, Dennis J McFarland, and Gert Pfurtscheller. Eeg-based communication: improved accuracy by response verification. *IEEE transactions on Rehabilitation Engineering*, 6(3):326–333, 1998.
- [WYK17] Nicholas R Waytowich, Yusuke Yamani, and Dean J Krusienski. Optimization of checkerboard spatial frequencies for steady-state visual evoked potential brain–computer interfaces. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 25(6):557–565, 2017.
- [XLX⁺13] Bin Xia, Xing Li, Hong Xie, Wenlu Yang, Jie Li, and Lianghua He. Asynchronous brain–computer interface based on steady-state visual-evoked potential. *Cognitive Computation*, 5(2):243–251, 2013.
- [XWP⁺18] Xiao Xing, Yijun Wang, Weihua Pei, Xuhong Guo, Zhiduo Liu, Fei Wang, Gege Ming, Hongze Zhao, Qiang Gui, and Hongda Chen. A high-speed ssvep-based bci using dry eeg electrodes. *Scientific reports*, 8(1):14708, 2018.
- [YCW⁺15] Peng Yuan, Xiaogang Chen, Yijun Wang, Xiaorong Gao, and Shangkai Gao. Enhancing performances of ssvep-based brain–computer interfaces via exploiting inter-subject information. *Journal of neural engineering*, 12(4):046006, 2015.
- [YLC⁺13] Chia-Lung Yeh, Po-Lei Lee, Wei-Ming Chen, Chun-Yen Chang, Yu-Te Wu, and Gong-Yau Lan. 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, 2013.

- [YLP⁺14] Lingling Yang, Howard Leung, David A Peterson, Terrence J Sejnowski, and Howard Poizner. Toward a semi-self-paced eeg brain computer interface: decoding initiation state from non-initiation state in dedicated time slots. *PloS one*, 9(2):e88915, 2014.
- [YXL⁺17] Wenqiang Yan, Guanghua Xu, Min Li, Jun Xie, Chengcheng Han, Sicong Zhang, Ailing Luo, and Chaoyang Chen. Steady-state motion visual evoked potential (ssmvpe) based on equal luminance colored enhancement. *PloS one*, 12(1):e0169642, 2017.
- [ZCFC13] Rosanne Zerafa, Tracey Camilleri, Owen Falzon, and Kenneth P Camilleri. Comparison of plain and checkerboard stimuli for brain computer interfaces based on steady state visual evoked potentials. In *Neural Engineering (NER), 2013 6th International IEEE/EMBS Conference on*, pages 33–36. IEEE, 2013.
- [ZGW08] Haihong Zhang, Cuntai Guan, and Chuanchu Wang. Asynchronous p300-based brain–computer interfaces: A computational approach with statistical models. *IEEE transactions on bio-medical engineering*, 55:1754–1763, 07 2008.
- [ZHK⁺13] Claudia Zickler, Sebastian Halder, Sonja C Kleih, Cornelia Herbert, and Andrea Kübler. Brain painting: usability testing according to the user-centered design in end users with severe motor paralysis. *Artificial intelligence in medicine*, 59(2):99–110, 2013.
- [ZKWR08] Thorsten Zander, C Kothe, S Welke, and Matthias Roetting. Enhancing human-machine systems with secondary input from passive brain-computer interfaces. *Proceedings of the 4th International BCI Workshop & Training Course*, pages 144–149, 01 2008.
- [ZTLZ17] Nannan Zhang, Jingsheng Tang, Yadong Liu, and Zongtan Zhou. An asynchronous ssvep-bci based on variance statistics of multivariate synchronization index. In *Biomedical Engineering International Conference (BMEiCON), 2017 10th*, pages 1–4. IEEE, 2017.
- [ZXCY14] Yangsong Zhang, Peng Xu, Kaiwen Cheng, and Dezhong Yao. Multivariate synchronization index for frequency recognition of ssvep-based brain–computer interface. *Journal of neuroscience methods*, 221:32–40, 2014.
- [ZXL⁺12] Yangsong Zhang, Peng Xu, Tiejun Liu, Jun Hu, Rui Zhang, and Dezhong Yao. Multiple frequencies sequential coding for ssvep-based brain-computer interface. *PloS one*, 7(3):e29519, 2012.