

RADBOUD UNIVERSITY NIJMEGEN

BACHELOR THESIS OF ARTIFICIAL INTELLIGENCE

Tagging the World

Broad-Band Noise Tagging as an Alternative to Steady-State Frequency Tagging

Author: J. Thielen

Supervisors: prof. dr. ir. P. Desain dr. J. Farquhar

July 29, 2013

Contents

1	Intr	roduction 4
	1.1	Brain Compute Interfaces
		1.1.1 Purpose
		1.1.2 Process
		1.1.3 Challenges
	1.2	Tagging the World 6
2	Bac	kground 7
	2.1	Frequency tagging
	2.2	Noise tagging
3	Ain	n 10
Δ	Mei	thods 11
•	4 1	Participants 11
	4.2	Stimuli
	4.3	Equipment 11
	4.4	Design \ldots 12
5	Δns	alvees 13
0	5.1	Pre-processing 13
	$5.1 \\ 5.2$	Two-class 13
	0.2	5.2.1 Linear Discriminant Analysis
		5.2.1.1 Kernal Logistic Regression
		5.2.2 Template Matching Classifier
		5.2.2.1 Dynamic Channel Selection
		5.2.2.2 Spatial filter methods
		5.2.2.2.1 PCA component of LDA classifier 15
		5.2.2.2.2 PCA component of ERP 15
		5.2.2.2.3 ICA component of trials
		5.2.2.2.4 CCA component of trials and ERP 16
	5.3	Reconvolution
	5.4	Multi-class
	5.5	Optimal subsets
6	\mathbf{Res}	ults 18
	6.1	Two-class
	6.2	Reconvolution
	6.3	Multi-class
	6.4	Optimal subsets
	6.5	Questionnaire
7	Dis	cussion 22
	7.1	Two-class
	7.2	Spatial filter methods 22
	7.3	Reconvolution method
	7.4	Multi-class
	7.5	Optimal subsets

	7.6 Design rules	$23 \\ 24$
8	Conclusion 3.1 Future work	26 26
9	Acknowledgements	27
10	References	28
11	Appendix A: Sequences	31
12	Appendix B: Reconvolution	36
13	Appendix C: Questionnaire	39

Abstract

In the field of EEG-based Brain Computer Interfaces the Evoked Potential is a well-studied response. To elicit this response, frequency tagging is a common paradigm to use. In this paradigm a periodic stimulus signal elicits Steady-State Evoked Potentials.

Recently, an alternative has been proposed called noise tagging. In this method Pseudo-Random Noise sequences are used to watermark stimuli. These Broad-Band stimuli also elicit Evoked Potentials in EEG. Like Steady-State Evoked Potentials they can be used in different sensory modalities, in this paper visual stimulation with LEDs was used. Using noise tagging it is possible to model and predict the response based on the specific stimulus bit-sequence. The paradigm that was developed allows for short train sessions, a high number of classes that require no extra classifier training, a prior selection of a set of optimal code sequences and high information transfer rates.

1 Introduction

1.1 Brain Compute Interfaces

Brain Computer Interfaces (BCI) are systems that enable users to communicate intentions or mental states to the outside world. This communication process is accomplished by using direct brain activity only. More specifically, the source is limited to the central nervous system, so no parts of the peripheral nervous system are used. That means muscles, eye movement, and nerve signals outside the skull are not used at all.

BCIs are implemented for many purposes. Examples are cursor-control (Trejo et al. (2006)), speller-devices (Treder & Blankertz (2010), Waal et al. (2012), Furdea et al. (2009)), wheelchair-control (Philips et al. (2007)), neuralrehabilitation (Daly & Wolpaw (2008)), error-detection (Blankertz et al. (2003)), gaming (Nijholt et al. (2009)), and many more. First, note that these BCI rely on different sensory modalities like the visual, auditory or tactile stimulation. These domains are both used to stimulate the user and therefore to elicit a response in the brain, as well as to give a certain feedback. Second, note that the brain signals are recorded using different kinds of neuroimaging techniques. Methods like Electro-Encephalography (EEG), Magneto-Encephalography (MEG) and functional Magnetic Resonance Imaging (fMRI) are non-invasive, meaning they are outside the skull. Methods like Electro-Corticography (ECoG) and single neuron recording are invasive and need surgery to be placed in or on the brain. Third, note that different tasks elicit different brain responses. For example, focused attention tasks typically evoke Event-Related Potentials (ERPs) or Evoked Potentials (EPs), whereas imagined-movement is based on Oscillatory signals like Event Related Synchronisation and Desynchronisation (ERS/ERD).

1.1.1 Purpose

From this point, a relevant question would be "why should one use a BCI?" For healthy persons this seems a good question, currently, because BCIs are still outperformed by normal Human Computer Interaction (HCI) devices (e.g., mice and keyboards). Furthermore, healthy people are a lot faster and still more accurate in performing tasks without a BCI intervening. That because a BCI requires time to process and reliability is often weak.

However, there is a large group of patients ranging from those with limited damage to motor-areas to patients who are completely paralysed. Especially in the later stages of Amyotrophic Lateral Sclerosis (ALS or Locked-in syndrome) patients are entirely incapable of any communication with the outside world. BCIs could make a difference to these patients, as they do not rely on the peripheral nervous system, but on brain activity itself.

1.1.2 Process

Then the next question would be "how does a BCI work?" The BCI-cycle (Gerven2009) provides four main stages in the process of a BCI, see Figure 1. These four main stages are visited each time a user tries to communicate one intention.

The first stage is called the encoding stage. In Figure 1 this is shown, presenting Stimulation, Modality, Task and User. In this stage the user translates his intentions according to a specific task via a specific modality, and according to a specific stimulation. So for example, the user has to focus (task) visually (modality) on a flashing screen (stimulation). This stage makes sure task relevant, or intention relevant brain signals are elicited.

In the second stage the elicited

brain responses are recorded. In Figure 1 this is shown by Measurement. As outlined above, brain activity could be measured by different devices, like fMRI, EEG and many more. In all cases, this stage measures the brain response that was elicited in the first stage.

In the third stage, which is called decoding, the recorded brain signals are interpreted. As shown in Figure 1, this involves three steps: Preprocessing, Feature Extraction and Prediction. During pre-processing the recorded brain signals are filtered from external noise and taskirrelevant brain activity like on-going



Figure 1: The BCI-cycle explaining the different stages of a BCI (Gerven et al. (2009)).

cognitive processes. After preprocessing, relevant features that distinguish the encoded intentions from each other are extracted, such that it is possible to analyse the signals. In the last step, these features are used to predict or detect the real intention of the user. Basically, this stage labels the incoming data with a label saying which intention is most likely to be coded by the measured brain signals.

The fourth stage, the transduction phase, provides the user with feedback on the decoded intention. In Figure 1 this is called Output. For example, if attending a specific area on a screen would mean you want to open a door, this stage causes the door to be opened. The output - as it can be perceived by the user - by itself constitutes a kind of stimulation. When this stimulation is equal to the intended outcome it can be interpreted by the user as a reward on a well-transmitted signal. This can trigger learning in the user, which may either help the BCI because the user adapts, or hinder because the signals become non-stationary. The latter may require the detection algorithm to adapt as well.

1.1.3 Challenges

So far, it seems a straightforward way to develop a BCI. However, this is not the case. There are various problems that BCIs have to overcome to achieve good performances. First of all low Signal-to-Noise Ratio (SNR) makes is difficult to interpret brain signals. Essentially this means that the brain recordings contain far more noise than it contains the desired signal. This noise is not only external noise, like 50 Hertz signal noise due to electrical grid fluctuations, but also task-irrelevant brain activity like on-going cognitive processes. In addition, muscle

movements typically evoke big responses in the brain (e.g., eyes, neck, heartbeat, etc.). Furthermore, if non-invasive techniques are used, the brain signals have to propagate through brain tissue, the scalp and skin before they reach electrodes. On the other hand, if invasive techniques are used, which typically give higher SNR, still tissue forming or neuroplasticity cause low SNRs.

Second there is subject-to-subject or inter-subject variability. In other words different persons elicit different brain response even under equal circumstances. This variability is not only found in the amplitude of the signal, but also in the spatial orientation of the response. In addition, even in the same person differences can be found over different time intervals, which is called session-to-session or inter-session variability. This variability could be caused by user-learning or habituation, but even in really small time intervals, already different brain activity is found.

Obviously, this raises questions like "what needs to be done to improve BCI?" (Desain, Farquhar, Haselager, et al. (2008)) Basically, there are two ways to improve BCIs. On the one hand the signals could be enhanced. On the other hand, detection and decoding could be enhanced. Considering the task of the brain, better stimulation paradigms are developed that evoke responses that are better distinguishable from the noise. Considering the task of the computer, either detection could be improved by having other or new neuroimaging techniques. Or decoding could be improved by having other or new analysis and machine learning techniques.

1.2 Tagging the World

Tagging the World is a study in which an EEG-based BCI application is developed using the visual modality. It is the aim to enable users to control objects scattered around in the near environment by just looking at them. Rapid flashing LED's are used to evoke brain patterns. Note, that each object in the environment is linked to a light, which is flashing with a specific pattern. For example, if there are five objects that could be manipulated, five lights are flashing, each with a different pattern. The response on such a pattern is detected using EEG recordings. By analysing the brain responses it is possible to retrieve which flash pattern, (and thus which object) the user was looking at. This object could then be manipulated. By manipulation one can think of a door opening or closing, a TV turning on or off, or even giving a moving Roomba a specific command.

The BCI that is used in Tagging the World could easily be adapted to be applicable for visual spellers, where each character is flashing with its own specific pattern. In fact, the lights could be arranged in different orientations, as long as they are perceivable. Therefore it is also possible to play games with this system. The system has already been applied on the can-toss game, demonstrating to a general public that it is possible to throw cans by just looking at them.

The aim of the research is proposing an alternative method to watermark these objects with visual flicker. Therefore, the research is dealing with the challenges of BCI addressing both systems involved: first the brain with its responses to new stimuli and second the computer with new analysis techniques.

2 Background

In the domain of EEG-based BCIs that use rapid visual flicker two paradigms are commonly used. On the one hand frequency tagging and on the other hand noise tagging. Both provide flash patterns that evoke specific responses in the brain.

2.1 Frequency tagging

Frequency tagging is the type of stimulus tagging in which each stimulus flashes with its own frequency. As often square-waves (on-off) are used, the frequency spectrum of one Frequency tag shows a peak at the specific fundamental frequency and all odd harmonics. In Figure 2 two Frequency tags are shown with their frequency spectrum.

Frequency tags elicit Steady-State Evoked Potentials (SSEP), which essentially means that the same frequency of the stimulation can be found back in the brain responses. To analyse these responses, one should thus look into the frequency domain. Next to the power at a specific frequency, phase can be used as feature as well, because phase coupling of stimulus and response is assumed. In that sense SSEP is different from the induced responses like alpha-power that are not coupled to external stimulation.

However, it is shown that each individual has his own sensitivity pro-

Figure 2: Two frequency tags with their corresponding frequency spectrum. Note that the top frequency tag was presented at 40 Hertz and the other at 80 hertz.

files for different frequencies (Bieger (2010)). This is the so called inter-subject variability. Another disadvantage of frequency tagging is the low noise robustness. Suppose there is noise over 40 to 60 Hertz, and the stimulation frequencies are 45 and 55 Hertz. Because the entire signal is covered by noise, there is a lower chance of detecting which stimulus was focused on. Moreover the brain emits spontaneous oscillatory signals in this range that vary with mental activity and thus will hinder reliable detection of frequency tags in the stimuli.

Consider a multi-class BCI system in which sixty classes are used. The frequency tags have to be above 40 Hertz, such that the flickering is invisible, but below 100 Hertz, because above it the signal is not perceivable anymore in EEG. So the system is left with classes that use frequencies that only differ 1 Hertz. This gives rise to high spectral overlap if reasonably short time windows are used and thus no clear ability to distinguish between the stimuli from brain signals. Therefore spectrally dense frequency tagging is not easily applicable to SSEP-based BCIs.

However, on the contrary, SSVEP-based BCI have proven to be very successful. Table 1 lists the five best SSVEP-based BCIs with their respective Information Transfer Rates (ITR) based on the Wolpaw ITR formulation (Kro-

negg et al. (2005)). The fastest SSVEP-based BCI transfers on average 68 bits per minute.

Authors	CR	С	T (sec)	ITR (bits/min)
Martinez et al. (2007)	0.97	4	3.5	30.0
Bin, Gao, Yan, et al. (2009)	0.85	6	na	39.7
Parini et al. (2009)	0.98	4	2.0	51.5
Bin, Gao, Wang, et al. (2009)	0.95	6	na	58.0
Gao et al. (2003)	0.88	48	3.8	68.0

SSVEP-based BCIs

Table 1: The five best SSVEP-based BCIs with their respective number of classes (C), proportion correctly classified performance as Classification Rate (CR), seconds per trial (T) and Information Transfer Rate ITR (bits/min). This table is adapted from (Vialatte et al. (2010)).

2.2 Noise tagging

Noise tagging is the type of stimulus tagging in which each stimulus flashes with its own pattern. Analysing such a signal gives rise to a broad-band spectrum. In other words, the frequency spectrum of a Noise code shows peaks over the entire spectrum. While the term Noise tags can be used for both analog noise and digital bit sequences, in this paper we focus on (up to a certain upper time-scale) non-periodic bit sequences. These sequences are usually generated with a shift register of a specific length M that feedbacks an xor of some of its outputs to its input. If the outputs are chosen well the register cycles through all its possible states (except the all zero state). The resulting sequence is called an m-sequence (Maximum length). Combining specific choices of two of these sequences in an xor yields a so called Gold code. These codes have special autocorrelation properties - it does not resemble itself at any time lag apart from 0. Time shifting one of the generating m-sequences by different delay times yields a family of $(2^M + 1)$ different Gold codes that also have a desirable non-correlation between any pair. These families are used in wireless telecom broadband applications and are very robust. Gold codes are almost balanced (same number of 0's and 1's) and exhibit a smaller number of occurrences of larger runs of the same symbol, up to a sequence of M 1's.

However these long runs are a problem for BCI stimuli because often brain responses are triggered by transitions and while flicker above a certain clock frequency is not very noticeable or annoying , it becomes like if it contains long on or off periods. Therefore the codes are modulated with twice its clockfrequency, a kind of phase-shift keying, which produces a very homogenous sequence with only runs of one or two 1's or 0's. This reduces the low frequency content of the broadband signal. Assuming that each (same-length) pulse in the sequence exhibits a same response, and all these components combine into the full response, linear regression can be used to derive these components. Using a method similar to convolution, the response to new sequences can also be predicted. This general model thus allows for zero-training of new classes. The modulated sequences (m-Gold codes) still exhibit the desirable auto- and crosscorrelation properties, but these properties do not necessarily carry over in the domain of EEG responses. However, as a way to model and predict responses for untested sequences is developed, also a search of a subset of sequences that are easily distinguishable in the responses can be carried out, assuming a way can be found to limit the combinatorial explosion for larger numbers of classes. The resulting optimal subset of codes are called Platinum codes. Section 11 lists the above mentioned Noise tags with their specific properties and generation process in detail.

In Figure 3 two Noise codes are shown. Figure 3 also shows the frequency spectrum of the two Noise codes. Noise tags elicit Broad-Band Evoked Potentials (BBEP), which essentially means that each pattern evokes its own response in the brain, according to the pattern itself. To analyse these responses, one needs to train on all patterns, such that by averaging templates so called Event-Related Potentials (ERP) can be constructed. Basically, one is then analysing in the time-domain.

Only since the paper by Sutter



Figure 3: Two noise tags with their corresponding frequency spectrum.

(1992) Pseudo-Random Noise sequences are applied in BCI. See Table 2 for an overview of the five best BBVEP-based BCIs. These BCIs all have in common that they use m-sequences as stimuli and Canonical Correlation Analysis (CCA) as classification method. Spüler et al. (2012a) achieved higher performances by also using Support Vector Machines (SVM), and in a later version they again improved their BCI with Error-Related Potentials (ErrP) (Spüler et al. (2012b)). Until now, the fastest BBVEP-based BCI transfers on average 144 bits per minute.

DD (Jour		
Authors	CR	\mathbf{C}	T (sec)	ITR (bits/min)
Bin, Gao, Wang, et al. (2009)	0.91	16	1.05	92.8
Sutter (1992)	na	64	1.20	100.0
Bin et al. (2011)	0.85	32	1.05	108.0
Spüler et al. (2012a)	0.96	32	1.05	133.6
Spüler et al. (2012b)	0.96	32	1.05	144.0

BBVEP-based BCIs

Table 2: The five best BBVEP-based BCIs with their respective number of classes (C), proportion correctly classified performance as Classification Rate (CR), seconds per trial (T) and Information Transfer Rate ITR (bits/min). This table is adapted from (Duijn (2012)).

3 Aim

The current research aims to propose a new stimulus tagging paradigm, namely using an Optimal Subset of Modulated Gold codes (OSMOG or Platinum codes) to watermark stimuli. As a baseline comparison frequency tags are used.

To investigate this new paradigm, several steps need to be made. The first research question is how we can classify or even how we can simply choose the class of which the template (ERP) maximally correlates with the single trial. The second research question focusses on the selection of an electrode or a combination of the signals of the different electrodes by a spatial filter to create one-dimensional templates. The third research question proposes a method called reconvolution that allows for predicting templates for new unseen sequences. The fourth research question will then investigate the ability to go multi-class, even up to sixty-five classes. The fifth research question aims to find a method to select optimal subsets of stimuli within these sixty-five classes.

Only for the first and second research question frequency tagging will be used as baseline for reference. All research questions are summarized in Table 3 with the stimulus tags and number of classes that are used.

#	Question	Tags	# Classes
1	Linear Discriminant and Correlation Analyses	f,n	2
2	Spatial filtering methods	f,n	2
3	Reconvolution method	n	2
4	Multi-class	n	265
5	Optimal subsets	n,p	265

Research Questions

Table 3: An overview of all research questions and whether they are applied on both Frequency (f) and Noise (n) tags or on Platinum (p) codes. In addition the number of classes the analyses are applied on is shown.

4 Methods

4.1 Participants

Three university students participated in the experiment. Two participants were male, one female, aged 21 to 25 years. They were free of any experience with epilepsy and had normal vision abilities (e.g., no glasses). They participated voluntarily and were not paid for their contribution.

4.2 Stimuli

The experiment presented two Frequency tags and sixty-five m-Gold codes. The two Frequency tags were at 40 Hertz and 80 Hertz. The m-Gold codes were generated as outlined in Section 11 with a register length M = 6, and a preferred pair of linear feedback tap positions at [6 5 2 1] and [6 1]. Thus the sequence length before modulation was $2^6 - 1 = 63$ bits. The modulated bit length was 126 bits and was presented with a 160 Hertz clock. Thus the longest and shortest period present in the noise signal and the only periods present in the respected frequency tag represent 40 and 80 Hertz. This makes the signals very similar, only differing in being periodic or not. Figure 4 shows the two frequency tags



Figure 4: Two frequency tags and three noise tags that are used in the experiment. The stimuli were presented using a clock at 160 Hertz.

and three noise tags from top to bottom at the left side and from left to right on the right side.

4.3 Equipment

The sequences were projected onto a white wall using a desktop light (IKEA KVART desktop light). The desktop light was adapted to use LEDs. To control the flashes a custom build Midi-interface was used. At any time an opto-coupler recorded the actual flashing patterns, as a separate channel on the EEG recording. As measured by this sensor the timing accuracy was always within 1.6 ms (std of 0.74 ms).

The EEG data was recorded using the Biosemi ActiveTwo amplifier and the 64-channel cap. This amplifier samples at a frequency of 2048 Hertz, though the data was immediately down-sampled to 640 Hertz. It was specifically 640 Hertz such that the sample-rate of the flashing patterns is an integer divisor of it.

4.4 Design

Participants were first asked to fill out an Informed Consent Form. During cap-montage participants were given task-relevant instructions about the experiment. The experiment was divided into eight blocks that all cover the same trials in random order. During each trial, the participant had to focus at the light presented on a wall approximately two meters in front. The participant had to push a button to start the trial. After pushing the button the light flashed a specific pattern for 3 seconds. Between flash sequences the light stayed on constantly.

During each block, the two frequency tags and two pre-selected noise tags were presented in 13 trials, and one trial of all other tags were presented. Each block thus contained 13 * 4 + 1 * 63 = 115 trials. In each trial, the desktop light flashed for three seconds starting at button press.

When all blocks were finished, the participant was asked to fill out a questionnaire (see Section 13). This questionnaire provides a notion whether the participant was aware of the different flash patterns, what these looked like and whether they were irritating. A typical question was "if you were to count the number of different flash patterns, how many were there?"

5 Analyses

5.1 Pre-processing

All data is pre-processed according to the following pipeline:

- 1. All trials are linearly de-trended.
- 2. Bad channels are detected and removed using Spherical Spline Interpolation (SSI).
- 3. Channels are re-referenced using Common Average Referencing (CAR).
- 4. Trials are spectrally filtered using two pass-bands at 10 to 48 Hertz and 52 to 100 hertz.
- 5. Trials that deviate more than 3.5 standard deviations from the tag-related mean are removed from the data set.

5.2 Two-class

This section fully exploits the collected train-data for two classes. Using 10-fold cross-validation overfitting is prevented for all methods. Classification rates expressed as a percentage of correct classifications will be the criterion on which success of a method is evaluated.

5.2.1 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is a pattern recognition method that separates different classes by finding a linear combination of features. It is therefore also used for dimensionality reduction, but it can also function as a classifier.

A LDA classifier works in a supervised way. During training the system requires at least two class data including the labels, otherwise it is not able to define the optimal combination of features that separate the classes from each other.

5.2.1.1 Kernal Logistic Regression A Regularised Kernel Logistic Regression Classifier (KLR) is used as classifier. Regularisation parameters are set using cross-validation. During training the system optimises a decision function that maps the data to the labels. The optimisation is achieved by least squares regression.

The train phase of KLR is straighforward:

1. Train the classifier on train data

The test phase is as follows:

- 1. Apply the classifier on test data
- 2. The sign of the result yields the labels

5.2.2 Template Matching Classifier

Template Matching is based on defining a template T for each class and a distance function between a single trial and each template. The template that is closest to the trial is chosen by the classifier and designates the assumed class of this single trial. A simple and commonly used template is the ERP obtained by averaging over all trials corresponding to a class according to the following formula:

$$T_k(t) = \frac{1}{N} \sum_{n=1}^{N} x_{k,n}(t)$$

, where $T_k(t)$ is the template for class k, N is the number of trials for class k and $x_{k,n}(t)$ is the *n*th trial of class k.

Once these templates are defined, unseen data can be classified by correlating it with all templates. Taking the best correlating template can be seen as the template that best fits the trial, and therefore yields the most probable label of the unseen data. The corresponding formula is as follows:

$$\arg\max(c_k) = \frac{T_k^{\mathsf{T}}(t)x(t)}{\sqrt{T_k^{\mathsf{T}}(t)T_k(t)\cdot x^{\mathsf{T}}(t)x(t)}}$$

, where $T_k(t)$ is the template for class k, x(t) is the unseen single trial and c_k is the resulting correlation with class k which is maximized.

A problem with the above mentioned classification algorithm by template matching is that is assumes data of one channel (i.e., one electrode). As is mentioned, the brain signals are measured over 64 electrodes. In the next sections different methods are discussed that reduce this dimensionality.

5.2.2.1 Dynamic Channel Selection In Dynamic Channel Selection (DCS) all channels are used separately in classifying a single trial and a measure is defined which channel to take as output. For template matching e.g. the channel with the least distance to the closest template is chosen as best reflecting the class maximizing over template fit and channels at the same time. This method is new and yields unexpectedly good results.

DCS works directly with test data that is not labeled and is therefore unsupervised. DCS reduces the dimensionality as follows:

- 1. Correlate all templates with the single trial, but now also for all channels
- 2. Select the best channels by maximizing over channels
- 3. Apply template matching

5.2.2.2 Spatial filter methods The previous single channel methods ignore much valuable information that is in other unselected electrodes. Using spatial filters it is possible to cancel out noise and extract the relevant information from all electrodes by combining them to one. In the following pipelines, the template matching classifier is still used to find the label of unseen data.

5.2.2.2.1 PCA component of LDA classifier A LDA classifier yields a time*channels weighting matrix. Dimensionality reduction by designing and applying a spatial filter can be done by applying Principal Component Analysis to these classifier weights (PCA-C). Time and spatial regularities are then effectively decoupled.

PCA tries to extract the linearly uncorrelated variables, the principal components, from the data. By selecting the first principal component, the component is selected that accounts for as much of the variability in the data as possible. In terms of applying PCA on classifier weights, the variable that is consistent over the learned features and therefore explains most of the variance over time is selected as a spatial filter.

Because this requires a classifier to be trained, it is a supervised process that requires at least two class data. The process of PCA-R is as follows:

- 1. Train a classifier
- 2. Apply PCA on the classifier weights
- 3. Select the first principal component as spatial filter
- 4. Apply the spatial filter to the data
- 5. Apply template matching

5.2.2.2 PCA component of ERP Principal Component Analysis can also be applied to a template. This is referred to as PCA on a response, thus PCA-R.

PCA tries to extract the linearly uncorrelated variables, the principal components, from the data. By selecting the first principal component, the component is selected that accounts for as much of the variability in the data as possible. In terms of applying PCA on an ERP, the variable that is consistent over time and therefore explains most of the variance over time is selected as a spatial filter. PCA-R tries to find any time consistent spatial patterns in a template.

Because PCA-R only requires one template it is unsupervised and suffices with one class data. Applying PCA-R can be done following:

- 1. Compute a template by averaging over the trials from the one-class data
- 2. Apply PCA on this tenplate
- 3. Select the first principal component as spatial filter
- 4. Apply the spatial filter to the data
- 5. Apply template matching

5.2.2.3 ICA component of trials Another way to reduce the dimensionality is applying Independent Component Analysis (ICA). ICA tries to extract the independent components by maximizing the statistical independence of the estimated components. The two most important definitions of independence are minimization of mutual information and maximization of non-Gaussianity. In terms of applying ICA to EEG data, ICA extracts independent components over time like heart-beat, breathing, eye artefacts, but also the desired signal.

ICA can be applied directly on the data and suffices with one class data. However, providing multiple classes could enhance the performance of ICA. Deriving an unmixing matrix with ICA itself is an unsupervised method and therefore, even if providing multi class data, does not require any labels. However, picking the best component is. This is often done by human eye. In this study, we use the calculated criterion to pick the component that survives timelocked averaging best and makes single trials most resemble the ERP of their class.

The process of ICA is as follows:

- 1. Apply ICA directly to the data
- 2. Dynamically select the component that when applied, maximizes correlation between trial and template
- 3. Apply template matching

5.2.2.2.4 CCA component of trials and ERP Canonical Correlation Analysis (CCA) optimizes the correlation between trials and the corresponding template by defining a weighting matrices W_x and W_y . CCA optimizes these weightings in such a way that if W_x is applied to the trials and W_y is applied to the template, correlation between them will be maximized. Selecting the first component from CCA yields the optimal weightings over X and T.

Because CCA only requires trials of one class, from which also the template can be constructed, CCA is an unsupervised method that suffices with one class data. The process of CCA is as follows:

- 1. Compute the template T by averaging over the trials from the one-class data X
- 2. Construct a matrix R that contains as many as templates as trials in X
- 3. Find the optimal weightings W_x and W_y as follows:

$$\arg\max(W_x, W_y) = \frac{W_x^{\mathsf{T}} X R^{\mathsf{T}} W_y}{\sqrt{W_x^{\mathsf{T}} X X^{\mathsf{T}} W_x \cdot W_y^{\mathsf{T}} R R^{\mathsf{T}} W_y}}$$

- 4. Select the first canonical component as spatial filter
- 5. Apply the spatial filter to the data
- 6. Apply template matching

5.3 Reconvolution

Consider the methods above that use a TMC. These methods all require full training which means that for each class a template has to be constructed by averaging over trials. However, if the number of classes grows, more trials have to be presented which can become inconvenient. When working in a domain in which the stimuli for each class and build form smaller repeating building blocks, and a model can be found to decompose the response to the whole stimulus into a combination of the responses to each basic component, there is a way out. In linear systems convolution can be used to (de)compose a response of a system to a summation of scaled and time-shifted responses to individual small pulses of which the input signal can be considered to be constituted. In our case a variant is used in which the components are not delta-pulses but small block pulses.

And by using (re)convolution, it turns out that measuring responses for all possible input stimuli is no longer necessary. Reconvolution provides a way to collect data of only one class and predict templates for all others. The algorithms behind reconvolution are outlined in Section 12.

By applying reconvolution only the TMC changes:

- 1. Construct a template T by averaging over all trials of one class data
- 2. Predict all templates by applying reconvolution
- 3. Apply one of the dimensionality reduction methods
- 4. Apply template matching

5.4 Multi-class

To test the applicability of reconvolution for multi-class purposes, the pipeline given in Section 5.2.2.2.4 is used on N = 2 up to N = 65 classes. For each N, ten random subsets are selected that are evaluated to investigate the performance on N classes.

5.5 Optimal subsets

To improve the applicability of reconvolution for multi-class purposes, the pipeline given in Section 5.2.2.2.4 is used again on N = 2 up to N = 65 classes. However, now for each N, the optimal subset is selected and evaluated. The optimal subset is found by finding the Least Correlating Subset (LCS) within the correlations between all 65 possible classes predicted by reconvolution. The algorithms used to find the LCS are outlined in Appendix A referring to Platinum codes.

6 Results

6.1 Two-class

For both frequency tagging and noise tagging all methods outlined in Section 5.2 are validated using 10-fold cross-validation on the train-sets. These train-sets consisted of two classes. For each class 50 trials were used with a trial length of 0.78 seconds. For the m-Gold codes this means that exactly one period is presented each trial. The classification rates with respect to Frequency tagging are shown in Table 4, the results with respect to Noise tagging are shown in Table 5.

Note that these tables show different classification methods in one table. More specifically, the tables can be grouped according to their respective classification domain (e.g. LDA classifying time signals and TMC matching signals), or for TMC even the spatial filters that are used.

6.2 Reconvolution

All methods outlined in Section 5.2 are also used with reconvolution. However, because reconvolution uses train data of only one class, the LDA and PCA-C method are not applicable. These two require supervised data of at least two classes. In addition, frequency tagging is left out from here, as reconvolution cannot be applied to frequency tags if only one class is used. The classification rates using reconvolution are shown in Table 6.

6.3 Multi-class

Figure 5 shows the performance against trial size in seconds and train size in number of trials for a 6-class problem using the CCA method as outlined in Section 5.2.2.2.4.

Hroanoney	torrance
rieuuenuv	Lagging

Subject	KLR	DCS	PCA-C	PCA-R	ICA	CCA
1	0.83	0.83	0.79	0.82	0.85	0.91
2	0.60	0.76	0.65	0.64	0.72	0.92
3	0.47	0.64	0.49	0.41	0.59	0.92
mean	0.63	0.74	0.64	0.62	0.72	0.92

Table 4: Classification rates using frequency tags and full-training on a two-class problem. The methods are validated using 10-fold cross-validation on 50 trials of each class. Each trial took 0.78 seconds of EEG data. The classification rates are given for different analysis methods that are outlined in section 4.5.2.

		ono uag	88			
Subject	KLR	DCS	PCA-C	PCA-R	ICA	CCA
1	0.96	1.00	0.99	0.99	0.99	1.00
2	0.93	0.95	0.91	0.89	0.89	0.98
3	0.85	0.92	0.96	0.55	0.80	1.00
mean	0.91	0.96	0.92	0.81	0.89	0.99

Noise tagging

Table 5: Classification rates using noise tags and full-training on a two-class problem. The methods are validated using 10-fold cross-validation on 50 trials of each class. Each trial took 0.78 seconds of EEG data. The classification rates are given for different analysis methods that are outlined in section 4.5.2.

D	1
Reconvo	liition
ICCCONVO	luulull

Subject	KLR	DCS	PCA-C	PCA-R	ICA	CCA
1	-	0.96	-	0.93	0.84	0.98
2	-	0.81	-	0.75	0.73	0.98
3	-	0.92	-	0.52	0.71	1.00
mean	-	0.90	-	0.73	0.76	0.99

Table 6: Classification rates using noise tags and reconvolution on a two-class problem. The methods are validated using 10-fold cross-validation on 50 trials of each class. Each trial took 0.78 seconds of EEG data. The classification rates are given for different analysis methods that are outlined in section 4.5.2.



Figure 5: The performance against trial size in seconds and train size in number of trials for the CCA method applied on a six-class classification problem. The performances are averaged across all three subjects.

In Figure 6 the performance is given using the CCA pipeline and optimal parameters, namely 0.78 second trials and 50 train samples. The performance is plotted against the number of classes. Also a classifier performing at chance level is plotted, and the corresponding classifier that performs exactly significant (alpha = 0.05) above chance level. On the right the corresponding Information Transfer Rate (ITR) is given as measured by the Wolpaw equation (Kronegg et al. (2005)).



Figure 6: Performance and Information Transfer Rate plotted against the number of classes used. On the left side trained on the one m-Gold code, on the right side trained on the other. For both a trial length of 0.78 seconds and a train size of 50 trials is used. Note the two lines representing the chance level and the significance level above chance with 0.05 confidence level.

6.4 Optimal subsets

Subset selection was performed using an incremental method that at each step added the least maximum correlating class to the current subset, until a subset of a given size was formed. Figure 7 presents the same data as used in Figure 6, but now using an optimal subset. Again also chance level and 0.05 significance level are plotted.



Figure 7: Performance and Information Transfer Rate plotted against the number of classes used by using the optimal subset. On the left side trained on the one m-Gold code, on the right side trained on the other. For both a trial length of 0.78 seconds and a train size of 50 trials is used. Note the two lines representing the chance level and the significance level above chance with 0.05 confidence level.

6.5 Questionnaire

Two of the three participants filled out the questionnaire. These two indicated that both participants were not consciously aware that they had focussed at 67 different flash patterns. Participants wrote they could distinguish approximately four different patterns, especially ascribing huge difference between the noise and the frequency codes. In addition, from the questionnaire the conclusion could be drawn that these participants rated the flickering as being moderately annoying, in one participant even resulting in a light feeling of a headache.

7 Discussion

7.1 Two-class

Section 5.2 shows the classification rates for both frequency tagging and noise tagging. From these it can be concluded that noise tagging outperforms frequency tagging in all methods. However, it is believed that the lower classification rates of frequency tagging are caused by improper stimuli selection. The two Frequency tags were 40 Hertz and 80 Hertz, which were chosen to have fair comparisons with the m-Gold codes as these are the building blocks of the m-Gold codes. However, these two frequencies show overlap in the higher harmonics, which could be of disadvantage in correlation measures and even in classifiers. It is therefore hypothesized that frequency tagging would perform better if proper stimuli were chosen. Therefore, the study cannot yet say anything about noise tagging performing better than frequency tagging.

7.2 Spatial filter methods

Section 5.2 shows the classification rates for both frequency tagging and noise tagging by using different methods. All spatial filtering techniques, except CCA, are still outperformed by DCS. DCS even outperforms KLR. Note that DCS only used Oz, O1 or O2 as best channels to classify on, which perfectly matches the visual stimulation. However, the CCA pipeline seems to be the best method to use. It is being hypothesized that this method performs best because it uses the same optimization criterion (correlation) that is used in the classification pipeline.

7.3 Reconvolution method

Section 5.3 discussed reconvolution applied to a two-class problem, which makes the comparison between full training and reconvolution easier. Reconvolution shows good performances, especially for the CCA pipeline. Therefore it could be stated that reconvolution is able to predict the responses well, such that classification rates do not decrease much.

Reconvolution is able to fit a response very well. However, this response can be varied in length, such that more data is used to fit a bigger response. The Figures below show the correlations of a two-class problem by using different response lengths. In particular it shows the correlation of the real ERPs with the predicted ERPs. The left figure represents reconvolution trained on class 1, the right figure represents reconvolution trained on class 2. First of all, it can be observed that using longer responses enhances correlation between the train class and its prediction. However, this can be seen as over fitting, because the correlation between the real and predicted ERPs of the non-train class decreases after 0.15 seconds. Furthermore, the cross-correlations stay low, making the distinction between auto and cross correlations big. In all, a response of 0.15 seconds fits the data best.



Figure 8: Correlations between real ERPs (r) and predicted ERPs (p) using reconvolution on a 2-class problem. On the x axis the response length in seconds is varied between 0.05 and 0.45 seconds. On the left side reconvolution is trained on class 1, on the right side it is trained on class 2. These figures represent the average correlation over all subjects.

7.4 Multi-class

Section 5.4 shows that even in high numbers of classes reconvolution together with CCA performs really good. Using low amounts of train data and short trial lengths, still good performances are achieved that result in high ITRs. However, there is something strange going on. Recall that subject 1 achieved highest performances in the two-class full-train section. The same subject performs worst using reconvolution in multi-class classification problems. Furthermore, the classification rates of different participants differ much. In addition training on the one train class yields different classification rates than training on the other. Again some inter-subject and also inter-stimulus variability is found which do not rely on bad channels, bad trials or bad experiment design.

7.5 Optimal subsets

Section 5.5 shows the performance and ITR by selecting the optimal subset instead of some random ones. The method used, incremental least subset selection, still gets stuck in local maxima, but is able to improve the classification rates and ITRs. Especially in the range between 20 to 40 classes, where the most possible combinations are possible, the classification rates increase as compared to random subset selection.

7.6 Design rules

From our experiment, given the specific modulated Gold codes, 160 Hertz clock, the design rules for a BCI with a required number of classes and classification rate can be made. The design rules specify the trial length and number of trails needed for training to achieve a specific given classification rate for a particular number of classes. These design rules are not yet completed as not enough data is obtained. However, given a number of classes, the number of train samples does not matter much. On the contrary, the trial size does have great influence on the classification rate. Only if a higher number of classes is used, the number of train samples becomes more important.

7.7 Neuroscientific interpretation

Apart from the classification rates, results may come out of this research that are relevant for fundamental insights in neurocognition. One fundamental issue is whether SSEP stimulation works because of attunement of neural oscillators that become frequency and phase-locked to external stimuli (Kelso (1995)). It is not easy to derive characteristics of these hypothesised oscillators, e.g the time course of their locking behaviour when a new stimulus appears and their tracking behaviour when stimuli change frequency or phase. This is because the measurement filters used in capturing their output also need reasonable amounts of stable signal before they can detect changes. However, computational models of assemblies of oscillators exist (Large & Kolen (1994) and Large & Palmer (2002)) and form a good background for conducting research on their characteristics.

In this paper we propose non-periodic stimulation. If the responses to these stimuli are predicted well by the models, and the same models predict the responses to periodic stimulation as well and predict them with the same spatial distributions, we can conclude that hypothesising an internal active oscillator that becomes coupled to the stimulation is not needed. A simple linear treatment of the stimulus is then the better model. This has many consequences when evaluating a large body of literature (Jensen et al. (2012), Kelso (2002), Kalisch et al. (2009) and Maris et al. (2013)).



Figure 9: Topoplots of PCA-R component of presented with a noise tag (left) and frequency tag (right) of one subject. Note that the colour range differs.



Figure 10: Topoplots of CCA Wx component of presented with a noise tag (left) and frequency tag (right) of one subject. Note that the colour range differs.

As shown in Figure 9 the spatial filters constructed by PCA-R are different for frequency tags as compared to noise tags. It is hypothesised that this is caused by frequency tags demanding more complex visual processes that lie in higher-order visual areas. However, strangely this effect is diminished by using CCA, see Figure 10.

Frequency tags can be decomposed, but not as well as noise tags. Figure 11 shows the different subresponses extracted by applying reconvolution on noise tags and frequency tags. Note the difference in response between noise tags and frequency tags. Using reconvolution, 0.8 correlation values are achieved between real ERPs and predicted ERPs. For frequency tags this value is around 0.5. Thus there is extra processing over and above the decomposition. Though, still after applying reconvolution the CCA spatial filters are the same for Frequency en Noise tags. Thus there is a common linear processing active in both.



Figure 11: Shows the different suberps found by applying reconvolution. On the top the two subresponses for the two subcomponents if trained on noise tags, below if trained on frequency tags with the same subcomponents.

The residues after reconvolution can be also be CCA'd. These are expected to map only to the higher brain centres. It is then hypothesised that Noise tags also contain this signal but much less strong (occasional periodic parts), but it maps to the same place. Thus, there is a way to separate processing. In all, this also provides evidence for oscillators

8 Conclusion

During this study different classification pipelines are tested for both Frequency tags and Noise tags. First it can be concluded that Canonical Correlation Analysis (CCA) provides ways to design spatial filters which optimally combine electrodes to maximize correlation between trials and templates. Using CCA and simple correlation measures high classification rates are achieved.

Second, it can be concluded that the reconvolution method enables training on one class while predicting all others. In all it can be stated that using m-Gold codes, CCA and reconvolution, a computationally simple pipeline is constructed that is able to classify well. This pipeline predicts well even with low amounts of train data reflected in the number of trials and the trial length.

Third, it could be concluded that using Platinum codes, that are a subset of m-Gold codes selected by taking the least correlating subset in the predicted responses, it is possible to further improve the pipeline enhancing classification rates.

8.1 Future work

Future work could further improve the pipeline by improving the subset selection algorithm that currently still gets stuck in local maxima using the incremental method or still needs to visit a high number of combinations. In addition reconvolution parameters could be optimised like the events that are used to fit a response to. Currently only the length of on-pulses is used, whereas off-pulses could also convey information.

Future work could also investigate the effects of different stimuli. Currently only desktop lights are used, which is not a convenient solution for a real Tagging the World application. Smaller or even single LEDs could be a solution, but could also lower the brain response.

Another practical issue is the use of spatially overlapping stimuli, like in a speller design. The current research studied responses on single stimuli exactly in the perceptual field. Using the speller design, different stimuli would be used simultaneously. Then the stimuli have overlap and selective attention will play a critical role in encoding the intention from the brain signals. Thus, future work should validate the applicability of this system using multiple spatially overlapping stimuli.

The neuroscientific interpretation of SSEP and BBEP could still be improved too. We have outlined a reasoning scheme to investigate the neural processes behind SSEP with respect to neural oscillators. This work continues. Some other questions that are still open are whether BBEP also works at other clock frequencies, whether these will then use similar components shapes, and what will happen when a loose clock is used? However, apart from that, still a complete model of the cognitive processes behind SSEP and BBEP is still demanded which could broaden the perspective on these questions.

9 Acknowledgements

First of all I express great gratitude to Peter Desain and Jason Farquhar, both my supervisors. First of all for giving me the opportunity to work on such an interesting and new research project, and second for all the support I got. Also, I want to thank the Cognitive AI Department including Marjolein van der Waal, Jeroen Geuze, Alex Brandmeyer, Loukianos Spyrou, Frank Grootjen and Thea Holla for sharing thoughts, helping with conducting experiments or even be participant, arrangements and providing me with great suggestions. In addition, special thanks for helping on the implementation and hardware to the Technical Support Group including Philip van den Broek, Norbert Hermesdorf, Pascal de Water and Mark van de Hei. Last but not least, I thank my fellow students for helping me throughout the process, especially Geertjan Jacobs for providing comments on the pre-final version.

10 References

- Bieger, J. (2010). Stimulation effects in SSVEP-based BCIs. Unpublished master's thesis, Radboud University, Nijmegen. (Supervision P. Desain and G.G. Molina)
- Bin, G., Gao, X., Wang, Y., Hong, B., & Gao, S. (2009). VEP-based braincomputer interfaces: time, frequency, and code modulations [research frontier]. Computational Intelligence Magazine, IEEE, 4(4), 22–26.
- Bin, G., Gao, X., Wang, Y., Li, Y., Hong, B., & Gao, S. (2011). A high-speed BCI based on code modulation VEP. *Journal of neural engineering*, 8(2), 025015.
- Bin, G., Gao, X., Yan, Z., Hong, B., & Gao, S. (2009). An online multi-channel SSVEP-based brain–computer interface using a canonical correlation analysis method. *Journal of neural engineering*, 6(4), 046002.
- Blankertz, N., Dornhege, G., Schafer, C., Krepki, R., Kohlmorgen, J., Muller, K., et al. (2003). Boosting bit rates and error detection for the classification of fast-paced motor commands based on single-trial EEG analysis. Neural Systems and Rehabilitation Engineering, IEEE Transactions on, 11(2), 127– 131.
- Blankespoor, J. (2008). Noise tagging as a new auditory BCI-paradigm: a pilot study. Unpublished master's thesis, Radboud University, Nijmegen. (Supervision J. Farquhar and P. Desain and S. Gielen)
- Daly, J., & Wolpaw, J. (2008). Brain-computer interfaces in neurological rehabilitation. The Lancet Neurology, 7(11), 1032–1043.
- Desain, P., Farquhar, J., Blankespoor, J., & Gielen, S. (2008). Detecting spread spectrum pseudo random noise tags in EEG/MEG using a structure-based decomposition. In Proceedings of the 4th Int. BCI Workshop and Training Course 2008. (Graz, Austria)
- Desain, P., Farquhar, J., Haselager, P., Hesse, C., & Schaefer, R. (2008). What BCI research needs. In Proceedings of ACM CHI 2008 Conference on Human Factors in Computing Systems. (Venice, Italy)
- Duijn, A. van. (2012). Spread spectrum techniques in BCI: A review of auditory and visual BCI-systems using continuous and binary noise tagged stimuli. Unpublished master's thesis, Utrecht University, Utrecht. (Supervision P. Desain and N.F.Ramsey)
- Farquhar, J., Blankespoor, J., Vlek, R., & Desain, P. (2008). Towards a noisetagging auditory BCI-paradigm. In Proceedings of the 4th Int. BCI Workshop and Training Course 2008, 50-55. (Graz, Austria)
- Fries, P., Reynolds, J., Rorie, A., & Desimone, R. (2001). Modulation of oscillatory neuronal synchronization by selective visual attention. *Science*, 291 (5508), 1560–1563.

- Furdea, A., Halder, S., Krusienski, D., Bross, D., Nijboer, F., Birbaumer, N., et al. (2009). An auditory oddball (p300) spelling system for brain-computer interfaces. *Psychophysiology*, 46(3), 617–625.
- Gao, X., Xu, D., Cheng, M., & Gao, S. (2003). A BCI-based environmental controller for the motion-disabled. Neural Systems and Rehabilitation Engineering, IEEE Transactions on, 11(2), 137–140.
- Gerven, M. van, Farquhar, J., Schaefer, R., Vlek, R., Geuze, J., Nijholt, A., et al. (2009). The brain-computer interface cycle. *Journal of Neural Engineering*, 6(4), 041001.
- Gold, R. (1967). Optimal binary sequences for spread spectrum multi-plexing. IEEE Transactions on Information Theory, 13, 619-621.
- Heiden, L. van den. (2008). Steady-state evoked potentials for brain computer interface. Unpublished master's thesis, Vrije University, Amsterdam. (Supervision R. Schaefer and P. Desain)
- Jensen, O., Bonnefond, M., & VanRullen, R. (2012). An oscillatory mechanism for prioritizing salient unattended stimuli. *Trends in cognitive sciences*, 16(4), 200–206.
- Kalisch, T., Tegenthoff, M., & Dinse, H. (2009). Sensory stimulation therapy. Frontiers in Neuroscience, 3, 96–97.
- Kelso, J. (1995). Dynamic patterns: The self organization of brain and behaviour. The MIT Press.
- Kelso, J. (2002). The complementary nature of coordination dynamics: Selforganization and agency. NONLINEAR PHENOMENA IN COMPLEX SYS-TEMS -MINSK-, 5(4), 364–371.
- Kronegg, J., Voloshynovskiy, S., & Pun, T. (2005). Analysis of bit-rate definitions for brain-computer interfaces. In *Csrea hci* (pp. 40–46).
- Large, E., & Kolen, J. (1994). Resonance and the perception of musical meter. Connection science, 6(2-3), 177–208.
- Large, E., & Palmer, C. (2002). Perceiving temporal regularity in music. Cognitive Science, 26(1), 1–37.
- Maris, E., Womelsdorf, T., Desimone, R., & Fries, P. (2013). Rhythmic neuronal synchronization in visual cortex entails spatial phase relation diversity that is modulated by stimulation and attention. *NeuroImage*.
- Martinez, P., Bakardjian, H., & Cichocki, A. (2007). Fully online multicommand brain-computer interface with visual neurofeedback using ssvep paradigm. *Computational intelligence and neuroscience*, 2007, 13–13.
- Meel, J. (1999). Spread spectrum (Tech. Rep.). De Nayer Instituut.
- Nijholt, A., Bos, D., & Reuderink, B. (2009). Turning shortcomings into challenges: Brain-computer interfaces for games. *Entertainment Computing*, 1(2), 85–94.

- Parini, S., Maggi, L., Turconi, A., & Andreoni, G. (2009). A robust and self-paced bci system based on a four class ssvep paradigm: algorithms and protocols for a high-transfer-rate direct brain communication. *Computational Intelligence and Neuroscience*, 2009.
- Philips, J., R Millan, J. del, Vanacker, G., Lew, E., Galán, F., P.W.Ferrez, et al. (2007). Adaptive shared control of a brain-actuated simulated wheelchair. In *Rehabilitation robotics, 2007. icorr 2007. ieee 10th international conference* on (pp. 408–414).
- Regan, D. (1977). Steady-state evoked potentials. J. Opt. Soc. Am., 11, 1475-1489.
- Spüler, M., Rosenstiel, W., & Bogdan, M. (2012a). One class SVM and canonical correlation analysis increase performance in a c-VEP based brain-computer interface (BCI). In *Proceedings of 20th european symposium on artificial neu*ral networks (esann 2012). bruges, belgium (pp. 103–108).
- Spüler, M., Rosenstiel, W., & Bogdan, M. (2012b). Online adaptation of a c-VEP brain-computer interface (BCI) based on error-related potentials and unsupervised learning. *PloS one*, 7(12), e51077.
- Sutter, E. (1992). The brain response interface: communication through visually-induced electrical brain responses. *Journal of Microcomputer Applications*, 15(1), 31–45.
- Treder, M., & Blankertz, B. (2010). Research (c)overt attention and visual speller design in an ERP-based brain-computer interface.
- Trejo, L., Rosipal, R., & Matthews, B. (2006). Brain-computer interfaces for 1-d and 2-d cursor control: designs using volitional control of the eeg spectrum or steady-state visual evoked potentials. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 14(2), 225–229.
- Vialatte, F., Maurice, M., Dauwels, J., & Cichicki, A. (2010). Steady-state visually evoked potentials: Focus on essential paradigms and future perspectives. *Progress in Neurobiology*, 90, 418-438.
- Waal, M. van der, Severens, M., Geuze, J., & Desain, P. (2012). Introducing the tactile speller: an erp-based brain-computer interface for communication. *Journal of Neural Engineering*, 9(4), 045002.

11 Appendix A: Sequences

Below the creation processes and properties of m-sequences, Gold codes, m-Gold codes and Platinum codes are given. This forms an overview of Gold (1967) and Meel (1999) that may be consulted for further details. Note that for some measurements (balance property and run-length distribution), the sequences are assumed to contain ones and zeros, but others (auto- and cross-correlations) are expressed easier if these symbols are remapped to +1 and -1 respectively.

Maximum Length Sequences

Creation

A maximum length sequence (MLS), or m-sequence, is generated by having a Linear Feedback Shift Register (LFSR) of length M, with certain feedback tap positions. This register can initially be chosen randomly, leaving out the all-zero option. A common way to initialize the register is filling it with ones only. Another initial register will only produce a shifted version of the m-sequence. At each loop, the bits that are in the register at the feedback tap positions are xor-ed (modulo-2 addition), see



Figure 12: The m-sequence generation process for M = 5 with feedback tap positions at [6,1]. The bits at the taps are xor-ed, the results is inputed into the register, which therefore shifts one bit and outputs a bit. This process is repeated until a maximum length sequence is created.

Figure 12. The result is used as output and also fed back into the register that therefore shifts one bit. This procedure is repeated until a length of $N = 2^M - 1$ is reached. The register may never be in a state with all-zero's (as it will stay there indefinitely), hence the minus one. If, during the process, the initial register state shows up earlier than $2^M - 1$, the feedback tap points are chosen incorrectly. The feedback tap positions should be connected according to a primitive polynomial. A primitive polynomial is said to be primitive if it cannot be factored (i.e. it is prime), and if it is a factor of (i.e. can evenly divide) $x^N + 1$, where $N = 2^M - 1$ (the length of the m-sequence). All primitive polynomials that have a degree equal to M are considered to be fine for m-sequence generation. Note, that all good sets of feedback tap positions thus contain an even number of positions.

Properties

First of all, m-sequences are almost balanced. More specifically, m-sequences contain one more 1 than 0's. This is since the all-zero state is never met, meaning there are $N_1 = 2^{M-1}$ ones and $N_0 = 2^{M-1} - 1$ zeros.

Second, m-sequences have a special run-length distribution. For all runs, meaning subsequences of only ones or zeros, one-half has a length of 1, onequarter has a length of 2, one-eighth has a length of 3, and so forth and so on.



Figure 13: Two m-sequences with their frequency spectrum.

In addition, for each run, there is an equal amount of runs of ones and of runs of zeros.

Third, the auto-correlation of an m-sequence is two valued. Obviously, the auto-correlation at time-shift 0 is 1. At all other time-shifts the auto-correlation is equal to $-\frac{1}{2^M-1}$. Basically, m-sequences have the auto-correlation property that best allows detection of the sequence when its phase is not available.

Last, the cross-correlation (between sequences) is not as well as desirable as their auto-correlation. So two different m-sequences may still be highly correlated.

Gold Codes

Creation

A set of Gold codes is generated by having two m-sequences that are generated using preferred pairs of feedback tap positions.

Preferred pair of m-sequences are generated by having a preferred pair of feedback tap positions for the same register length M. A preferred pair of feedback tap positions satisfies the following conditions:

1. M is odd or mod(M, 4) = 2



Figure 14: The Gold code generation process: two m-sequences from which one is shifted in time, xor-ed to create a Gold code. This process is repeated by shifting one m-sequence over all time shifts, therefore creating all Gold codes.

2. take integer k and an odd integer q, than either $q = w^k + 1$ or q = 2(2k) - 2k + 1



Figure 15: Two Gold codes with their frequency spectrum.

3. the greatest common divisor of M and k, gcd(M,k) = 1 when M is odd and gcd(M,k) = 2 when mod(n,4) = 2.

If a preferred-pair of m-sequences is obtained, a set of Gold codes is generated by xor-ing (modulo-2 addition) the two at all time-shifts, see Figure 14. Because it is just adding, the products will also have a length of $N = 2^M - 1$. Because all time-shifts are used, $2^M - 1$ new codes can be generated, plus the two original m-sequences which gives a set of $2^M + 1$ Gold codes.

Properties

First of all, the balance property of Gold codes is not entirely met. Approximately one-third of a set of Gold codes is not balanced in which some contain more ones and some contain more zeros. For the two-third part that is balanced, all codes contain one more one than zeros, like an m-sequence has.

Second, the auto-correlation function of Gold codes is not as good as for m-sequences. Obviously, it shows a peak at 1 at time-shift 0. At all other timeshifts it shows the same three values as in the cross-correlations, but in different proportions.

At last, the cross-correlation of a set of Gold codes generated by a preferred pair of m-sequences is three valued. Recall that $N = 2^M - 1$. Then if M is even, the cross-correlation values are $\frac{-1}{N}$, $\frac{-2^{(l+2)/2)+1}}{N}$ and $\frac{2^{(l+2)/2)-1}}{N}$ with an occurrence of approximately 0.5, 0.25 and 0.25 respectively. If M is odd, then the cross-correlation values are $\frac{-1}{N}$, $\frac{-2^{(l+1)/2)+1}}{N}$ and $\frac{2^{(l+1)/2)-1}}{N}$ with an occurrence of approximately 0.75, 0.125 and 0.125 respectively.

m-Gold Codes

Creation

Modulated-Gold codes or m-Gold codes are generated by having a Gold code that xor-ed (modulo-2 addition) with a double-frequency bit-clock. This procedure is known as Phase Shift keying (PSK). Because the code is up-sampled, the length of the m-Gold code is $N = 2 * (2^M - 1)$.

Properties

All m-Gold codes are balanced. More specifically, each m-Gold code contains as much ones as zeros, thus $N_1 = N_0 = 2^M - 1$.

The normalized auto- and crosscorrelation properties of Gold-codes are retained by this modulation treatment.



Figure 16: The Modulated-Gold code generation process. A Gold code is modulated by xoring it with a double-frequency bit-clock.

The modulated gold codes only exhibit run-length of 1 or 2. This shapes their spectrum, and reduces low frequency content. This is a desirable property when used to modulate audio or light intensity as long runs become noticeable and annoying and make the tags perceptually distinguishable.



Figure 17: Two Modulated Gold codes with their frequency spectrum.

Platinum Codes

Given a generative model, a mapping from stimulus to response, one can consider correlations in brain activity to be able to take the P best performing

m-Gold codes. These ${\cal P}$ m-Gold codes are then considered to be Platinum codes.

Creation

Thus, Platinum codes are a subset of m-Gold codes. Assume a set of m-Gold codes is used to evoke brain responses, denote these responses as the set R. The set Platinum codes of P m-Gold codes, is that subset taken from R of size P that has the lowest maximum cross-correlation between any pair. This subset is called the Least Correlating Subset (LCS) and forms a set of Platinum codes.

The first and simplest way to find the LCS is by using a brute-force or exhaustive approach. This approach generates all possible subsets from R and computes the maximum cross-correlation within each subset. The optimal subset is then found by taking the argmin. This method will always find the optimal subset, but explodes in the number of combinations and is therefore not applicable.

Another method is finding the LCS by clustering the elements in R on similarity. The clustering is achieved by constructing a dendrogram using the hierarchical complete-link clustering algorithm. Afterward the tree is cut into P clusters. The search space is now narrowed down as from each cluster only one element has to be chosen. From this point, all possible subsets are created using one element from each cluster. The optimal subset is than selected by taking the argmin of the maximum between-pair correlations. However, as this method narrows down the search space, the number of combinations is still enormous making the evaluation computationally hard.

Another way to find the LCS is by stepwise selection. This method increases the subset by initially taking the least correlating pair from R. From that point it adds the element from R that has the lowest maximum correlation with the generated subset from the previous step. This step is repeated until the subset contains P elements. An extension to the algorithm is to evaluate the current subset also with backward stepping leaving out the worst correlating element in the current subset, and adding one from R that correlates be better. However, still this algorithm could get stuck in a local maximum as it evaluates step by step, looking one in front and one in behind. On the contrary, this method does find a solution in low amounts of time.

12 Appendix B: Reconvolution

In commonly used Brain Computer Interfaces (BCI), all present classes have to be seen in a training phase. However, when the number of classes gets bigger, like for example in a visual speller with twenty-six or even more characters, the training phase gets too large. Reconvolution is a way that enables to train on one class only, and predict all others. In the next sections, reconvolution will be explained in more detail.

Assumptions

Linearity

Reconvolution assumes the brain to be a linear system. So, first it should hold that if A is put into the system resulting in C, and afterward B is put into the system which results into D, inputting A + B should give C + D. In addition, it should hold that if the input is vA + wB the system should give vC + wD. Though it is known that the brain is not an entirely linear system as in the combination in EEG of contributions from different areas through volume conduction. And, as is shown in this thesis, reconvolution is capable of predicting a large percentage of the variance in the responses.

Composability

Reconvolution assumes that all classes are build up from the same building blocks. In fact, the class on which is trained, should be build up from all subparts that can be found in the to be predicted classes. In other words, reconvolution decomposes the trained class into subcomponents. For each subcomponent a response is found. Only these responses are used to build the predictions. Concluding, classes should have a similar composable structure. The components could be the actual value at each sample time (like in true convolution), but also the positive and negative flanks, or positive pulses of various lengths present in the signal.

Method

In Figure 18 the process of reconvolution is shown for the trained class only. On the left side, the composability is shown. On top the whole sequence is given, which is decomposable into two subcomponents, being a long and a short impulse, the events. These events are shifted in time such that, if all are added up, the original sequence is found back. During training by repetitive stimulation with this sequence, and eventually averaging over trials an ERP is constructed. The ERP corresponding to the sequence on the left is shown on the top right. It is then assumed that each subcomponent on the left, evokes its own subERP on the right. Note that for each short event, the same subERP is given and for each long one another. Summarizing the previous, we have a sequence that is decomposable into two subcomponents being a short and a long event. In addition we have an ERP corresponding to that sequence with its subERPs corresponding to these short and long impulses.



Figure 18: The process of reconvolution. On the left the decomposition and composition of a noise code. With arrows to the right the corresponding responses are shown. Following the same decomposition and composition process, reconvolution is able to predict responses.

Now, notice that also the subERPs are shifted according to the time-lag in the corresponding event. If all the subERPs are added up, it is assumed that the original ERP will be reconstructed. By applying least squares regression it is possible to find these specific subERPs on each event. By using a similar method as convolution, a prediction of the whole ERP can be constructed. However, due to the non-linearity of the brain this prediction is not entirely equal to the original ERP, but the normalized correlation between the two is on average 0.8.

Now note that all our sequences are build up from the same subcomponents, again recall the composability assumption. Because the responses on these subcomponents are already known, it is possible to fit these subERPs into the structures of all these other sequences.

Concluding, by training on one sequence, an ERP is obtained. Then, the sequence is decomposed into a structure giving the exact timings of the individual events. Using regression the ERP is also decomposed into the individual responses, yielding single responses for each individual subcomponent. Because all sequences (the other classes) are build up from the same subcomponents, the brain response can be predicted by convoluting the structure of these sequences with the subERPs.

Formal

The description above introduced the reader into the workings of the method. However, depicting it mathematically makes it more precise. In Desain, Farquhar, Blankespoor, & Gielen (2008) a formal description has been given on reconvolution using some other events as above. In the description rising (0-1) and falling (1-0) events are used, which would correspond to an on-going light and an off-going light respectively, in the context of this thesis. However, in this thesis short (1-1) and long (1-1-1-1) events are used. Note that it is thus possible to have different events. Another option could be to use both on and off sequences, including the short and long bits. The formal model that is presented below abstracts from these events, by providing a summation over all events, whatever they are or represent.

The model of decomposition could then be written as:

$$x(t) = \sum_{\tau=1}^{L} \sum_{j=1}^{E} I_j(t) r_j(t-\tau)$$

where, x(t) is the total response at time t, L is the duration of the response, E are the events, r_j is the response on the *j*th event and I_j is the indicator function that is 1 if the *j*th event is at time t and 0 otherwise.

To do the regression part a Toeplitz structure is used, that lists the events at each point in time. In Desain, Farquhar, Blankespoor, & Gielen (2008) the model is then expressed as:

$$x = \begin{bmatrix} \vdots & & \vdots \\ I_j(i:i+L) & \cdots & I_E(i:i+L) \\ \vdots & & \vdots \end{bmatrix} \begin{bmatrix} j \\ \vdots \\ r_E \end{bmatrix} = Mp$$

where, x are the modelled responses at each time, M is the decomposition matrix containing events shifted one sample at each row, and p is the concatenation of all the types of responses. This means that the equation is linear with respect to the responses, and as of that the parameters can be found using least-squares regression.

Optimal Component Selection

Throughout this article, multiple methods are outlined that find different components (e.g., PCA, ICA, CCA, but also DCS). Using reconvolution a new way to find the best component is acquired. Before reconvolution already a real ERP is defined by averaging over trials. By applying reconvolution a prediction of this ERP can be constructed. Call these ERP and PRD respectively. Now, by applying the component (e.g., select a channel or apply a spatial filter) to both ERP and PRD a correlation value can be computed corresponding to this component. By maximizing over this correlation, the best component is found. More specifically, by using this optimal component selection algorithm, that component is selected, that maximizes the correlation between the real ERP and predicted ERP.

13 Appendix C: Questionnaire

The questionnaire that is used to investigate the usability and user satisfaction of the tags that are used.

Tagging the World: Brain Computer Interface Noise-Tagging experiment

Radboud University Nijmegen

Questionnaire

ubject id:	
ubject id: _	

Welcome to this survey, which is about your experience in the BCI experiment you just participated in. Please fill it out as correctly as possible, so that your feedback can be used to improve the system. *Thank you for participating.*

About the experiment

- 1. During each block, have you perceived any differences in the flash patterns? If so, what difference?
- 2. If you were to count the number of different flash patterns, how many were there?
- 3. Explain yourself:
- 4. How well did you like the experiment in general? like O-O-O dislike
- 5. Explain yourself:
- 6. How well did you like the lights (flickering)? like _____ dislike
- 7. Explain yourself: