

An EEG- and ERP-based Image Ranking Application

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Medizinische Informatik

eingereicht von

Patrick Adelberger

Matrikelnummer 1128024

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung: Univ.Prof. Dipl.-Ing. Dr. Christian Breiteneder
Mitwirkung: Christoph Guger (g.tec)
Martin Walchshofer (g.tec)

Wien, 23.01.2019

(Unterschrift Verfasser)

(Unterschrift Betreuung)

An EEG- and ERP-based Image Ranking Application

MASTER'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Medical Informatics

by

Patrick Adelberger

Registration Number 1128024

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: Univ.Prof. Dipl.-Ing. Dr. Christian Breiteneder
Assistance: Christoph Guger (g.tec)
Martin Walchshofer (g.tec)

Vienna, 23.01.2019

(Signature of Author)

(Signature of Advisor)

Erklärung zur Verfassung der Arbeit

Patrick Adelberger
Mühlbachstraße 7a, 4451 Garsten

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

(Ort, Datum)

(Unterschrift Verfasser)

Danksagung

Hier möchte ich mich bei den Personen bedanken, die mich während der Umsetzung der Diplomarbeit motiviert haben. Im Besonderen bedanke ich mich bei meiner Familie, die mir immer zur Seite standen und mich unterstützt haben.

Außerdem möchte ich dem Unternehmen g.tec, in Schiedlberg, danken, für die tolle Zusammenarbeit. Vor allem dem Geschäftsführer Christoph Guger und dem Mitarbeiter Martin Walchshofer.

Abstract

An electroencephalogram (EEG) is a measurement to record the electrical potentials in the brain, also referred to as brain activity. It is widely used to study brain functions for neurological disorders. But for some time now, researchers use this EEG signal to create a brain-computer interfaces (BCI), which allows users to manipulate the computer or communication devices just with their thoughts. The P300 wave is a component of the EEG signal which is generated through an oddball paradigm. This oddball paradigm occurs if an infrequent target stimulus is mixed within a frequent non-target stimulus (visual or auditory stimuli). This P300 wave is often used in spelling applications for physically handicapped users where it is used in combination with the oddball paradigm, the P300-Speller.

The image ranking application in this thesis is based on a modified version of the P300-Speller for visual stimuli. This application ranks images of a picture set via their elicited responses in the EEG signal. The goal of this thesis is to check the functionality of the image ranking application with 2 different BCI devices and optimize the application run for image ranking.

The first part includes criteria about signal processing, e.g. sample rate. Here, the timing in which the EEG signal is sampled and sent to the application is analysed. After, the reliability of the recorded EEG signal the interstimulus interval (ISI) can be optimized.

The ISI consists of three parameters: the time an image is displayed on the screen, the time between two images and the number of times each image (flashes) has to be shown. These three parameters have to be tuned in a way that the accuracy is increased and the time for one application run is decreased.

Additionally, a ranking with different subjects should be created to depict if certain images are always ranked in the first few positions and are independent of the subject.

The results show that the interaction between application and the two BCI devices works as expected, short of some minor issues. Thus, the applicability of the application was given and the tuning of the ISI parameters could be started.

For the ISI the best accuracy was achieved when the images were displayed on the screen for 100 ms and 75 ms was used between two images. Furthermore, each image was flashed 20 or 5 times for the classifier or ranking, respectively. For the group rankings, the results indicated, that images of faces rank in average higher than the rest. Finally, the median of all rankings for one image should be used as the ranking parameter.

Keywords: EEG, BCI, Event-Related Potential, P300, Interstimulus Interval

Kurzfassung

Ein Elektroenzephalogramm (EEG) ist eine Messung zur Erfassung der Gehirnpotentiale oder auch Gehirnaktivität. Ursprünglich wurde es verwendet um neurologische Störungen zu untersuchen, allerdings wird es schon länger von Forschern genutzt, mit Hilfe von Brain-Computer Interfaces (BCI), Computer oder Kommunikationsgeräte zu steuern. Die P300-Kurve ist ein Signal, welches durch das Oddball-Paradignum erzeugt wird. Ein Oddball-Paradigma bedeutet, dass ein seltener Ziel-Reiz zwischen häufigen Nichtziel-Reize gemischt wird (visuelle und auditive Reize). Diese P300-Kurve wird häufig in Rechtschreibanwendungen für körperlich behinderte Benutzer verwendet, daher P300-Speller.

Die Anwendung für das Bilderranking in dieser Arbeit basiert auf einer modifizierten Version des P300-Spellers für visuelle Stimuli. Die Reihung der Bilder eines Bildsatzes wird anhand der Reaktionen im EEG-Signal berechnet. Das Ziel dieser Arbeit ist es, die Funktionalität der Anwendung mit 2 verschiedenen BCI-Geräten zu überprüfen und die Anwendungslaufzeit für das Ranking zu optimieren.

Die Funtionalitätsprüfung enthält Kriterien der Signalverarbeitung, z.B. Abtastfrequenz. Dabei werden die Zeitpunkte der Abtastung analysiert. Nachdem die Zuverlässigkeit des EEG-Signals geprüft wurde, kann das Interstimulusintervall (ISI) optimiert werden. Das ISI besteht aus drei Parametern: die Dauer der Anzeige eines Bildes auf dem Bildschirm, die Zeit zwischen zwei Bildern und die Anzahl der Wiederholungen (Flashes) pro Bild. Diese drei Parameter müssen so optimiert werden, dass die Genauigkeit erhöht und die Anwendungslaufzeit minimiert wird. Zusätzlich soll ein Ranking mit verschiedenen Probanden erstellt werden, um darzustellen, ob bestimmte Bilder immer in den vorderen Positionen gereiht werden und somit Unabhängigkeit von den Probanden sind.

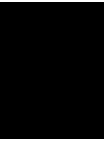
Die Ergebnisse zeigen, dass die Interaktion zwischen der Anwendung und den beiden BCI-Geräten wie erwartet funktioniert, abgesehen von einigen kleinen Problemen. Somit konnte mit der Optimierung der ISI-Parameter begonnen werden. Für das ISI wurde die beste Genauigkeit erreicht, wenn die Bilder für 100 ms auf dem Bildschirm angezeigt wurde mit 75 ms Pause dazwischen. Außerdem wurde jedes Bild 20 oder 5 Mal für die Klassifizierung oder das Ranking angezeigt. Bei den Gruppenrankings gaben die Ergebnisse an, dass Bilder von Gesichtern durchschnittlich besser gereiht werden als der Rest. Schließlich sollte der Median aller Rangfolgen für ein Bild als Parameter für die Bestimmung der Rangfolge verwendet werden

Keywords: EEG, BCI, Event-Related Potential, P300, Interstimulus Interval

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Aim of work	2
1.3	Methodological Approach	3
1.4	Structure	3
2	Background	5
2.1	Electroencephalogram	5
2.2	Classification	9
2.3	Application	10
2.4	Device	10
2.5	Setup	12
2.6	Signal Processing	12
3	State of the Art	15
3.1	Brain-Computer Interface	15
3.2	P300-Speller / P300-BCI	16
3.3	Interstimulus Interval	18
4	Experiments	21
4.1	Sampling Rate	21
4.2	Trigger Display	22
4.3	Trigger Frame	23
4.4	Set-up	24
4.5	Picture Sets	25
4.6	Copy Spelling	26
4.7	Ranking	27
4.8	Group Ranking	28
5	Results	35
5.1	Sampling Rate	35
5.2	Trigger Display	38
5.3	Trigger Frame	40

5.4	Picture Set	44
5.5	Copy Spelling	44
5.6	Ranking	46
5.7	Group Ranking	49
6	Discussion	55
6.1	Sampling Rate	55
6.2	Trigger Display	56
6.3	Trigger Frame	56
6.4	Picture Set	56
6.5	Copy Spelling	57
6.6	Ranking	57
6.7	Group Ranking	58
7	Conclusions	59
	List of Figures	61
	List of Tables	63
	Bibliography	65



Introduction

1.1 Motivation

The electroencephalogram (EEG) is a measurement of the brain activity and is widely used in science to study brain functions [27]. The basic principle to measure the brain activity is by means of a cap fitted with electrodes, which record the currents produced by brain cells [25]. Devices which record EEG signals and use them as an input for some application are called brain-computer interfaces (BCI).

There are diverse components in the EEG signal which record different functionalities of the brain itself. Even the placement of the electrodes is of importance, e.g. for visual processing the brain uses mainly the rear part.

If an infrequent stimulus (visual or auditory) is mixed within a frequent one the brain interprets this as an oddball paradigm. The P300 component identifies such a behaviour, this component is one of the most studied waveforms [4]. The name P300 or P3 is derived from the component's characteristics, as discussed in section Event-Related Potential. In fact, the name comes from [28], where the P300 was first discovered and peaked at 300 ms. However, the peak typically appears between 350 ms and 600 ms, but is still labelled P300 [19].

There are several fields of applications for the P300 component, one typical use case is where it is used as a spelling device (P300 Speller paradigm) [15], [31], [14], [20], [12].

A modified version of this use case is the basis of the image ranking applications developed by the company *g.tec*¹.

This application uses a set of pictures which will be shown in a fast sequence and measures the responding EEG signal to each image. Depending on these responses a ranking can be created for this picture set. This means a ranking is always relative to the picture set. It is important to keep in mind, that not only is the content of the image important but also the image properties (colour, contrast, brightness and so forth).

It should be mentioned that the type of reaction, to a picture, is not always the same. The appli-

¹<http://www.gtec.at/>

cation can only measure the response to a picture, but not the subjects feelings and/or emotions that may occur. Many different emotions can be associated with the response, e.g. happiness, disgust, lust, sadness, etc. Therefore, a simple statement to the ranking of different people, where the most attractive one is ranked first is wrong.

A commercial usage for this application could lie within the advertisement sector, where potential marketing images could be compared and ranked. The result of this ranking could then be used as an additional aspect for the decision making process. Especially, in the neuromarketing sector could this application be a major contributor.

The application was developed mainly for two devices:

- *g.USBamp* is a high end product and used in medical research, life science and biofeedback/BCI research.
- *g.Unicorn*'s domain lies in the non-medical environment for non-medical applications, e.g. games, and it is the low budget solution.

The applications usability has to be optimized to be commercially successful in regards to the result correctness and the run-time for one measurement. This means the interaction between the devices and the application has to be proper and the configuration parameters have to be refined. These parameters are:

- the display duration of each image
- how many image repetition are needed
- the length of the time between to images

This interplay of characteristics is called the Interstimulus Interval (ISI). The optimization of the ISI will reduce the run-time of the application, which would make one run more comfortable and lead to a higher readiness to use this application.

1.2 Aim of work

The basis of the application is the *P300 Speller* paradigm, which is normally used in combination with a BCI device to write on the computer. This modification leads to several different problems. Firstly, all the signal processing steps have to be tested and evaluated to be able to be used commercially. This includes the following tasks:

- functionality check of the set-up, interaction of the devices with the application
- application run optimization for the image ranking

The timings of the ISI parameters have to be checked, e.g. the time an image is depicted on the screen equals the configure time in the application. Furthermore, the timing in which the EEG signal is sampled by the device matches the defined parameter in the application.

Secondly, for the optimal configuration of the *Interstimulus Interval* in combination with the image ranking process, no papers were found. Hence, the basis for the optimization process

were papers on the P300 Speller paradigm. The three configuration parameters for the ISI are, *Flash Time*, *Dark Time* and *Number of Flashes*.

The Flash Time refers to the duration an image will be displayed for the user. Dark Time mean the opposite, the time between images were nothing but a black screen is shown.

Because of some uncertainties in the EEG signal, which is normal, and includes moving to much, talking, or lack of concentration, each images has to be depicted several times. This number is called the number of Flashes (images flash on the screen). These parameters have to be tuned to be as accurately as possible and keep the run-time to the minimum. This tuning of the parameters will take quite a significant amount of time, because of the high amount of combinations for the configurations.

Additionally, ranking of multiple runs should be created to determine if one of the images can really induce a stronger response. The right parameter for such a ranking has to be defined to negate some results deviating from the norm.

Furthermore, this ranking should be done with more than one person determine if the results of different subject are comparable.

1.3 Methodological Approach

In this thesis all analysis was done with *MATLAB*², which is a tool developed by *MathWorks*³. *MATLAB* combines a desktop environment with a matrix based programming language. For the image ranking application another *MathWorks* product was used, *Simulink*⁴ is a model-based simulation tool with real-time functionality. Extending the application with additional functions, needed to analyse the behaviour was rather easy, because it is based on *Simulink*.

All calculations were done offline, which means only the data of the recordings as well as the results of the application were used. Furthermore, *MATLAB* provides a vast array of existing functions, which were used instead of scripting them. These functions include calculating the mean, the median and the normal distribution of an array or read text files as well as generating plots. Therefore, mathematical formulas, such as mean, are not listed in this thesis, because of their simplicity.

Section Experiments presents a detailed description for all the assignments given for this thesis.

1.4 Structure

The following section State of the Art gives an overview of the current expertise in the topics regarding the applications configuration. It is divided into three subsections, the first part contains some examples for a brain-computer interface. The next part shows the P300-Speller applications and possibilities. Lastly, an overview of the interstimulus interval and its configurations are discussed.

²<https://de.mathworks.com/products/matlab.html>

³<https://de.mathworks.com/>

⁴<https://de.mathworks.com/products/simulink.html>

The methods and hypotheses, which were used for the achievement of the goals, defined in this master thesis, are identified and explained in section Experiments. It consists of three main parts. Firstly, the different experiments for the application in combination with the different devices. Secondly, the configuration of the different parameters to increase usability and accuracy. Lastly, the group rankings and the possible parameters to use for the rankings are explained. Additionally, two subjects were used to increase variations in the outcomes.

Furthermore, the outcome for all tests and methods demonstrated in section Experiments will be shown in section Results. The results will be discussed, explained, and compared to the current field of research in section Discussion.

Finally, the thesis ends with the conclusion of the work done in section Conclusions. It features some thoughts for possible future work as well as some difficulties which occurred during the composition of this paper.

Background

2.1 Electroencephalogram

Electroencephalogram or short EEG is an easy obtainable measurement, which records the electrical potentials in the brain. Therefore it is widely used in science to study brain functions and neurological disorders [27]. The basis for the EEG signal is the activation of brain cells (neurons) and the produced currents in them [25].

A neuron consists of a cell body, in which its nucleus is resident and processes branches also referred to as dendrites, see Figure 2.1. Moreover, information is transmitted via the axon, which is usually the longest dendrite [16].

As mentioned in [25], an action potential (*AP*) is the basis of the information transmission in nerve cells. Every action potential is created due to changes in the membrane potential of a neuron. Initialisation of the AP starts in the cell body and is transmitted along the axon. Generally speaking, the process to generate an action potential starts with the depolarisation (becomes more positive) of the membrane potential, which leads to a spike. The repolarisation (becomes more negative) of the neuron membrane potential occurs after its peak value. Furthermore, the potential after repolarisation is even lower than its resting potential and gets normal shortly after (Figure 2.2). In addition, the duration of this whole process takes five to ten milliseconds.

The detailed process about how the action potentials work on a molecular level can be found in [22]. Action potentials and their provoked postsynaptic potentials (*PSP*) are the fundamental parts for this task. There are two types of PSP: the excitatory postsynaptic potential (*EPSP*) and the inhibitory postsynaptic potential (*IPSP*). The connections between neurons are referred to as *synapses* and connect the end part of the axon of a neuron to the dendrites or cell body of another neuron. Every AP received creates a PSP in the neuron, that is either an EPSP or an IPSP for an excitatory synapse or an inhibitory synapse, respectively. Subsequently, the summation of these PSPs generate an action potential, if the threshold value is exceeded, see Figure 2.3.

Pyramidal neurons are common class of neurons found in the brain and they make up about 66% of all neurons in the cerebral cortex [2]. In fact, the currents that flow in these neurons during synaptic excitations are the measurable EEG signal [25]. Generally speaking, The sum-

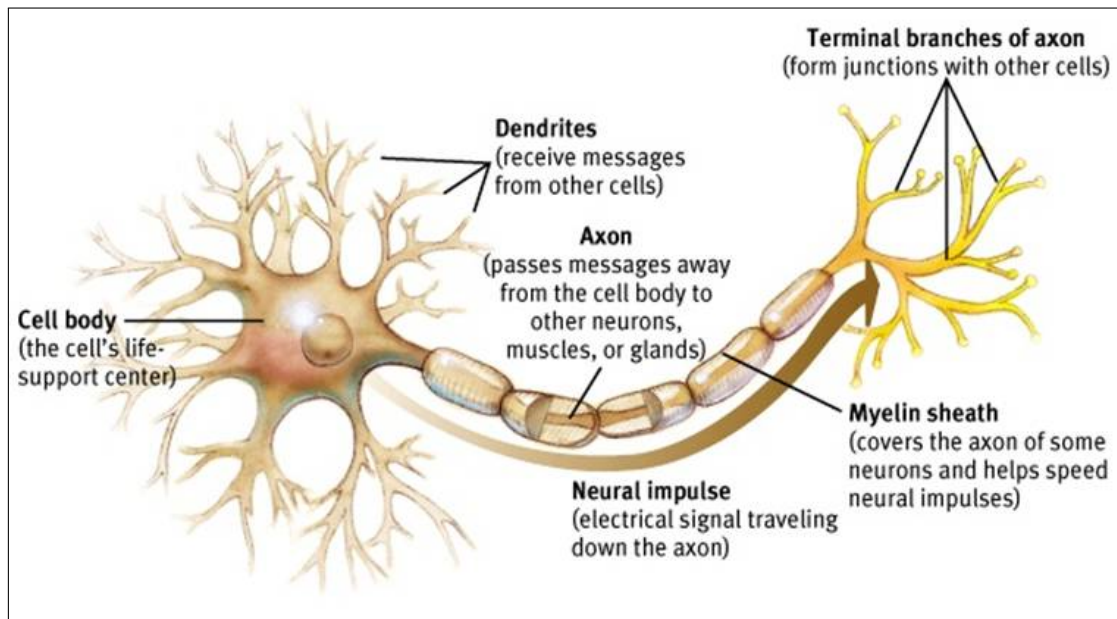


Figure 2.1: Structure of a neuron [1].

mation of inhibitory and excitatory PSPs in the cortex extend to the scalp surface where they are recorded. Additionally, a recordable activity on the scalp is only possible, if a large number of neurons are active [27].

EEG signals have an effective bandwidth up to 100 Hz and amplitudes from $5 \mu\text{V}$ to $100 \mu\text{V}$ [25]. Moreover, the frequency is split into five different bands, these bands represent the five major brain waves: *delta*, δ (0.5-4 Hz); *theta*, θ (4-8 Hz); *alpha*, α (8-13 Hz); *beta*, β (13-30 Hz) and *gamma*, γ (>30 Hz) [27].

Event-Related Potential

Event-related potentials (*ERPs*) are widely used for brain-computer interfaces and diagnostic purposes in psychiatry and neurology [24]. These ERPs are related to events, that are either internal or external, and can be elicited through stimuli, responses or decisions [19]. Consequently, an ERP is the summation of a large number of action potentials and creates voltage variations in the EEG signal [24].

In general, to obtain the ERP signal from the EEG, a particular stimulus or stimulus condition has to be presented repeatedly. Then the time locked EEG responses have to be averaged to eliminate the random part of the EEG signal. Thus, the part of the EEG signal, which is time locked to the stimulus remains and can be identified [6].

Moreover, the generated event-related potential waveform consists of a series of peaks and valleys called components, see Figure 2.4 [6].

Each component is labelled according to its polarity and either order of occurrence or latency. For instance, consider the P3/P300 wave, its peak is positive (P) at about 300 ms (P300) and it

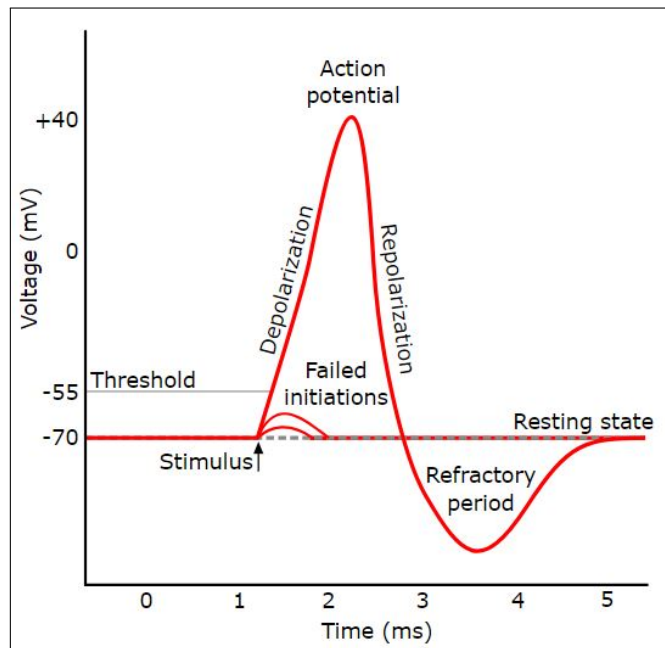


Figure 2.2: An example of an action potential [30].

occurs as the third component (P3) after the onset of a stimuli [4], [6]. However, the labelling of the components is not always consistent, because any component will produce a positive potential on one side of the head and a negative one on the other side [19]. In contrast, to the background EEG the ERP components are rather small with an amplitude from 1 to 30 μV [24].

Visual Evoked Potentials

As mentioned in [6], visual ERPs are changes in the EEG, that are elicited via a visual event such as the appearance of a picture on a computer monitor. Whereas, the visual evoked potentials (VEPs) are reflecting the basic visual functions (e.g. resolution, colour detection, motion detection) of a repeated visual stimulus without any cognitive content and task. A typical VEP occurs when a subject looks passively at a series of high contrast grayscale chequerboards. Generally speaking, the two potentials are frequently seen to be separate from each other, which is not entirely true, because perception and cognition aren't separate either.

P300/P3 Wave

The name P300 or P3 is derived from the component's characteristics, as discussed in section 2.1. In fact, the name comes from [28], where the P300 was first discovered and peaked at 300 ms. However, the peak typically appears between 350 ms and 600 ms, but is still labelled P300 [19].

Commonly, an oddball paradigm is used to generate the P300, where an auditory or visual stimulus is applied. Generally speaking, an infrequent target stimulus is mixed within a fre-

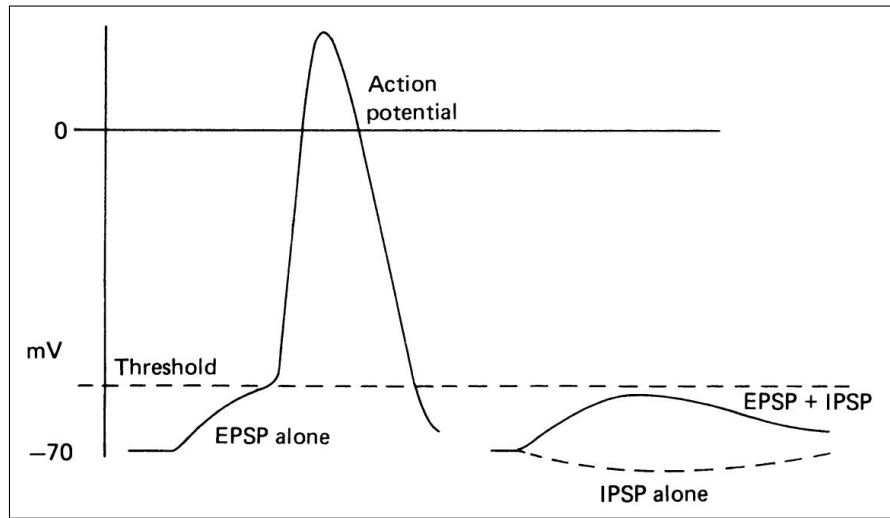


Figure 2.3: Membrane potential with the influences of IPSPs and EPSPs [17].

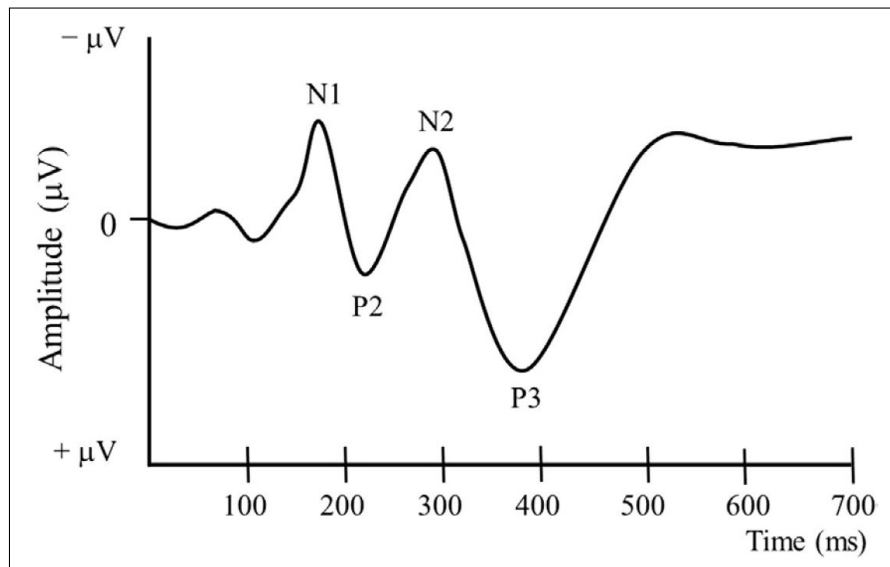


Figure 2.4: ERP with its components [4].

quent non-target stimulus, also known as *deviant* and *standard*, respectively [4]. In addition, the deviant is also referred to as *oddball* [19]. The subjects have to identify the infrequent target stimulus from the non-target stimulus and perform some kind of task, such as counting the occurrences. In Figure 2.5 the two different event-related potential waveforms for the oddball and standard target.

Furthermore, the P3 consists of two different subcomponents, *P3a* also referred to as novelty P3 and *P3b*. In addition, the localisation of these two parts is different, the P3a is located in

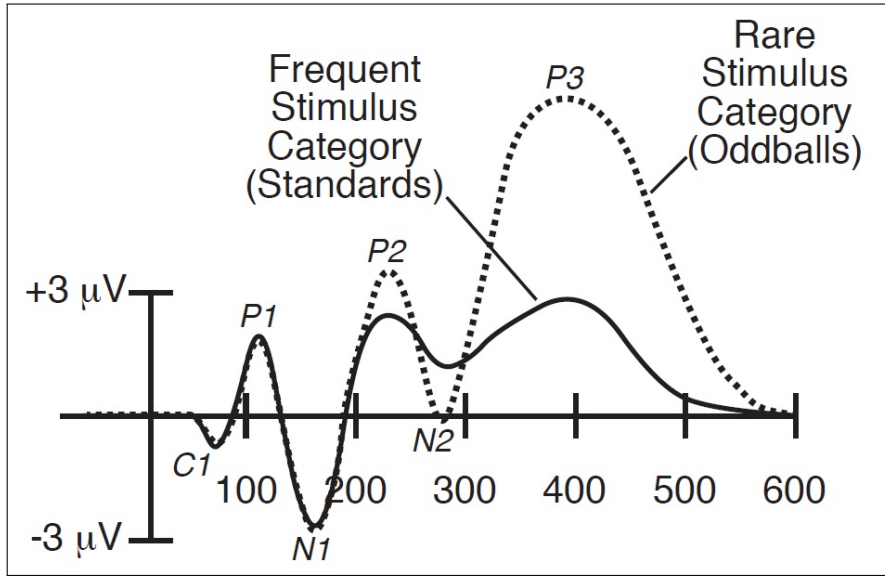


Figure 2.5: ERP waveform in a visual oddball paradigm [19].

the anterior scalp while P3b in the posterior scalp with its largest responses over the central and parietal area [19], [9]. P3a is elicited, as the name implies, by novel or salient stimuli independent of the task [24]. In contrast, the P3b is produced by the task given to the subject, on appearance of the oddball stimuli [19], [9].

2.2 Classification

The goal of classification is to predict the class, also known as category, of a set of given parameter values also referred to as *features* [5]. Furthermore, the process that allows the classifier to learn to recognise the category of a feature set is called training [3]. Usually, the training set consists of objects, each object has a set of features and is labeled with a class label. Then, in the classification process the algorithm analyses the objects and creates a description of all the classes defined in the training set [3]. Subsequently, the now trained classifier is able to detect the pattern of a new object and sort it into a category. There are many different types of classifiers, e.g., naive Bayes, decision tree, random forests, k-nearest neighbour, support vector machines, linear discriminant analysis, and so on [5], [3].

Linear Discriminant Analysis

Linear Discriminant Analysis (*LDA*) is a common technique to reduce dimensionality of features, the resulting lower dimensional features are much easier to handle [29]. Moreover, the LDA with a two class problem, which is called *Fisher's linear discriminant*, was later extended to more classes [23].

Generally speaking, the goal is to maximise the separation between the two means of the projected class, the *between-class variance*. Additionally, to minimise the variance, the *within-class variance*, of each projected class [29].

2.3 Application

The application runs as MATLAB Simulink simulation and has several different configuration options. The most important ones are the flash time, dark time, number of flashes and the running mode. Interstimulus interval (ISI) consists of two parts: the flash time (duration of trigger image presentation) and the dark time (duration of a black screen), see equation 2.1.

$$ISI = t_{Flash} + t_{Dark} \quad (2.1)$$

It also has two running modes *Copy Spelling* and *Ranking*, in copy spelling the target image is specified and the user has to focus on that one. For the ranking the user has to focus on all images with the same intensity. However, the application has a settling time of the filters of about 20 to 30 seconds, this time period will not produce a feasible EEG signal.

In both cases an LDA classifier has to be created which means starting the application in the copy spelling mode and set the parameters for flash time, dark time and number of flashes. When the software starts a window will pop up, where the whole picture set is displayed. The user has to choose one or more target images, with a left mouse-click on the picture. Afterwards, the process can be started and the current target image will be shown for three seconds. If more than one target image was chosen, then each target image will be displayed.

Secondly, when the copy spelling for classifier creation is done, a MATLAB script, provided by g.tec, calculates the parameters of the LDA classifier. This script computes the needed bias and the feature vector for the LDA classifier.

After a successful classifier was created, it is tested in the copy spelling mode. Again the user defines the target images and starts the copy spelling process. Furthermore, the accuracy of the current run will be displayed as a percentage value. A good classifier should have more than 80% accuracy. If the accuracy is lower another classifier should be created.

When both the classifier creation and the testing of this classifier are done and have a good accuracy the application can be started in the ranking mode. Another picture set will be used and the user concentrates on all the flashing images. The process of ranking has to be stopped manually and will then display a window with the ranking of the images in the picture set. Moreover, the application will create a text file with the ranking results.

2.4 Device

Both brain-computer interfaces were developed and are distributed by gtec¹. The difference between these two is that one of them is a high end product, *g.USBamp*, while the other one is a low budget solution, *g.Unicorn*. The *g.USBamp* (see Figure 2.7a) needs further hardware to get the EEG signal, in this case the *g.GAMMAbox* and the *g.GAMMAcap* to get the signal from

¹<http://www.gtec.at/>

g.Unicorn

The *g.Unicorn* is a biopotential amplifier with passive wet or dry electrodes. The data will be transferred wirelessly via the Bluetooth technology. It offers eight channels and each has a resolution of 24 bits. The sampling frequency is fixed to 250 Hz. The *g.Unicorn* is intended for a non-medical environment for non-medical applications and can be used for development, in the gaming industry or for the arts.

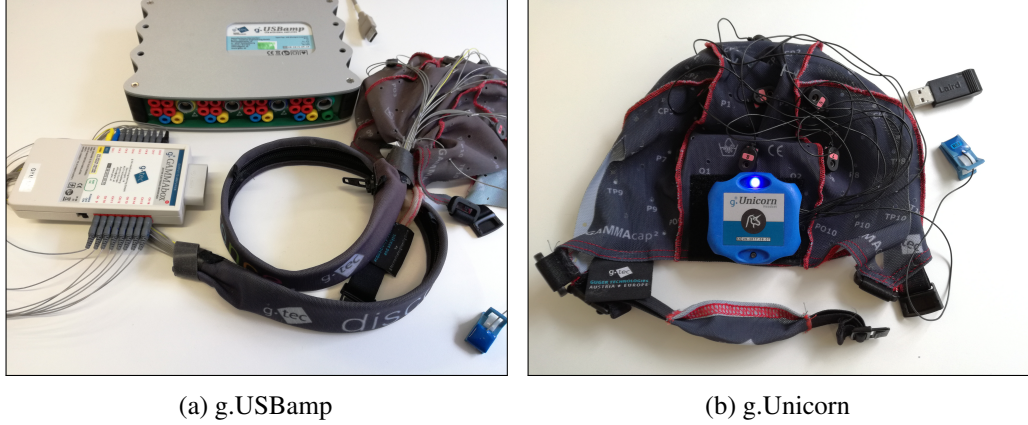


Figure 2.7: BCI devices used for this thesis.

2.5 Setup

A user can choose between the *g.USBamp* or the *g.Unicorn* configuration.

For the *g.USBamp* version the *g.GAMMACap* will be put on, this cap contains the electrodes for the signal capturing and it is connected to the *g.GAMMABox* which is an adaptor to connect with the *g.USBamp*. This *g.USBamp* is then connected via a USB cable to the computer.

The *g.Unicorn* setup has a similar cap as the other version, but on this cap the device is mounted. Here instead of an adaptor a USB Bluetooth dongle is plugged into the USB port on the computer.

On the computer the application runs as a MATLAB Simulink simulations and can be started when the devices are connected to the computer

2.6 Signal Processing

The EEG signal is sampled with 250 or 256 Hz, depending on the BCI device, *g.Unicorn* or *g.USBamp*, respectively, with a 24 bit value range. The used power supply has its own frequency of 50 Hz. This frequency could induce interferences in the EEG signal, therefore, a notch filter is applied to eliminate this possible influences. A notch filter is a special form of a band-stop filter. The speciality of this filter is that only a narrow range of frequencies will be filtered out.

After the notch filter a band-pass filter is applied, with a lower cut-off frequency of 0.1 Hz and a upper cut-off frequency of 30 Hz. The next step in the processing chain is to downsample the recorded data with a factor four. The normal behaviour of a downsampler is to use every fourth value of the data, in this case an average value of all four consecutive values is calculated and used instead of every fourth value.

Each image in the picture set has an identification value assigned to them, which is used to link the stimulus to the created EEG response. These responses are cut out of the EEG signal and used for the ranking. Start point of the response frames is from 100 ms before the stimulus to 700 ms after the stimulus. A mean value will be calculated from the first 100 ms, the pre-trigger samples. This values are used for the baseline correction of the EEG trigger response. Afterwards a moving average filter with a window size of 3 samples is applied. The last step is using a downsampler with the factor three, here every third value is used. All these values were predefined by the company

Each trigger calculates its feature vector based on the mean post-trigger responses of all the responses of the same stimulus. These feature vectors create a feature matrix, which is used in either creating a LDA classifier or to generate the ranking results.

State of the Art

3.1 Brain-Computer Interface

Brain-computer interface (BCI) technology was first developed to help people, which have trouble to communicate as well as restricted movement of body parts. These tasks are realised via different patterns of activity in the brain. The brain activity can be measured via two types of sensors, either outside the head, such as an electrode cap that detects the electroencephalogram (EEG), or sensors inside the head, which recognise the electrocorticography (ECoG) activity [11].

Examples for the usage of BCI system would be the selecting of letters via flickering icons on the monitor or the imagination of an extremity that induces movements for a mouse cursor or a wheelchair [11].

Furthermore, additional applications for a BCI systems were developed, which are also used in different areas, e.g. help mapping brain regions, recovery for stroke patients or to control the environment [11], [8].

A further development for future communication possibilities is presented in [13]. The study used the ECoG of patients, which were undergoing surgery for epilepsy and audio files were recorded to generate *Brain-to-Text* functionality. The subjects had to read different literature aloud and the recorded data was used to time-align phones with the ECoG signal.

For each phone, feature vectors were created from 200 ms before to 50 ms after a phone onset with multiple windows. These feature vectors were used in combinations with a pronunciation dictionary (list of the make up of words in phones) and speech recognition technology. The results show that with a dictionary size of 10 words, the correctly identified words were at over 60%.

Additionally, the phone correctness was analysed to circumvent the possibility of word recognition via a small group of phones, which has an average true positive rate of more than 20%. These findings support the hypotheses of the possibility to communicate with the means of brain activity. Moreover, with better models, the Brain-to-Text method allows the user continuous speech by only imagining to speak.

The control possibilities with and in an environment are shown in [8], which was simulated with the help of virtual reality. For this research three subjects performed a series of task within a virtual smart home. Each of the subjects had to perform a training on a BCI systems with a speller paradigm.

In fact, two different versions of the speller paradigm were used, the first was the single character speller (highlighting of one character) and the second one is the row/column speller (highlighting of one row or one column). Furthermore, every subject had to train with 42 characters and 15 flashes, which amounts to a training time of about 40 minutes. Afterwards, the BCI system was connected to the virtual reality (VR) system, where a virtual 3D smart home was running.

In this VR several different control elements were available to switch on and off the light, to open and close the doors and windows and so on. Overall seven different control masks with symbols were developed. The study shows that the used P300 BCI systems had an accuracy of 83% to 100% and that not every character had to be trained. Moreover, a higher amount of controls in a panel resulted in a higher accuracy, because the lower symbol possibility yields a higher P300 response. Additionally, the usage of symbols allows for a more goal-oriented implementation, like “move to this place” and the actual action is executed via a smart wheelchair.

3.2 P300-Speller / P300-BCI

The most used brain-computer interface is the ERP-based spelling device, often referred to as the *P300-Speller* or *P300-BCI* [15], [31], [14], [20], [12].

This P300-Speller is often used for patients with impaired motor functions but without cognitive handicaps [15], [20]. The classical P300-Speller is based on a 6 x 6 matrix, where the letters from A to Z and the digits from 0 to 9 are displayed. A subject focuses on a specific character they try to spell, the rows and columns of the matrix are flashed consecutively in a random order, this is called the *Row-Column (RC)* paradigm [15], [31].

Many studies and researches focus on improving this classical RC approach, in some cases the flashing pattern of the matrix is changed ([31], [14]) and in others the flashing illustration itself ([15], [31], [12]).

There are additional options to increase the accuracy of the P300-Speller [15], [31], [14], [20], [12]. The simplest solution is a higher number of receptions to average for the triggers, which has a negative influence on user experience because of the time extension. A further method is the modification of the interstimulus interval, which consists of the flash time and the dark time. All this mentioned approaches to improve the P300-Speller can also be combined to a certain degree.

The classical RC paradigm was compared with a random set-based stimulus pattern (RASP) for a 6 x 6 matrix [31]. In study [14] four different patterns for a 12 x 7 matrix were introduced. The parameters for this comparison is the accuracy in which a letter has been selected and the information transmission rate, which describes how fast information can be transmitted.

For the RASP, random characters flashed 12 times under the condition that each letter flashes twice [31]. Additionally, the four patterns for the 12 by 7 matrix were generated with 9, 12, 14 and 16 flashes, where each character flashes 3, 3, 3, and 2 times, respectively. For the RC approach 19 flashes were used [14]. Furthermore, the interstimulus interval for the flash times were 125 ms [31] and 100 ms [14] and the dark times 35 ms and 75 ms, respectively.

Generally speaking, the results of both, [31] and [14], show a higher accuracy and information transmission rate. The accuracy of the RASP was much better than the RC paradigm, because the combination of highlighted letters were shuffled each time and fewer number of flashes were needed to generate a level of 70% accuracy.

The four other flash patterns (9,12,14 and 16 flashes) performed compared to the RC (19 flashes) in terms of accuracy rather poorly, only the 16 flashes pattern had about the same accuracy than the 19 flashes pattern. The 9, 12 and 14 flashes pattern had a lower accuracy, but the information transmission rate was in all four patterns better. This outcome is based on the assumption that for the information transmission rate a accuracy level of bigger than 70% is sufficient.

Several studies show that changing the flashing illustration for the classical approach as well as new flashing patterns improve accuracy. Especially, modifying the flash depiction with a superimposed image of a face or replacing the character for the flash duration with an image of a face [15], [31], [12].

In [12] a 10 x 5 matrix (black background and white font) with an RC paradigm was used and the four different flashing animations accuracies were compared. The normal classical approach only highlighted the row or column by colour inversion, the three others replaced all characters in a row or column with images of faces. For one approach all faces were of Einstein in grayscale, another method was the same image of Einstein but in colour and the last flashing animation changed each character in the row or column into an image of a famous person, different persons were used.

A similar approach was used in [15] were the flash animation of a 6 by 6 matrix of a column or row was either a picture of a face of Albert Einstein or Ernesto “Che” Guevara. In addition, the pictures of a random reorganisation of all pixels of the images of Einstein and Che were used.

Moreover, the RASP pattern of [31] was used and extended with pictures of faces superimposed on the flashing characters (RASP-F). Furthermore, the used faces were generated by capturing the faces of the participants, all images had the same lightning and the hair was removed from the facial images. These facial images were either images of the subject, characters in a row, or images of friends or unfamiliar persons, characters in column.

For all these approaches different flash and dark times were defined, for [12] flash/dark time were 100 ms/75 ms, for [15] 33,3 ms/100 ms and in [31] 125 ms/35 ms.

Across the board all results using the face highlighting animations were better concerning accuracy, in comparison to the classic RC paradigm. For the faces used in [15] the time needed to spell a word could be reduced by 1,8. Furthermore, the coloured face of Einstein and of famous persons and the grayscale images of Einstein had all an accuracy of over 90% compared to the over 80% for the colour conversion flash animation [12].

The mention RASP-F outperformed the RASP approach, it was 2,3 times faster than the RC and 1,7 times faster than the RASP [31].

Furthermore, examination of the ERP to the face stimuli for the different approaches, like different paradigms of flash and dark time, showed that some components in the ERP were elevated, especially the N170 and the N400f [15], [31]. Additionally, the image of the face of the participant compared to the images of the face of friends or unfamiliar people shows a higher amplitude for the P300 and also the N400f components [31].

P300-Speller focuses on the languages with an alphabetic writing system (e.g. English and Ger-

man), but for logographic languages such as Chinese the P300-BCI is not a suitable solution [20]. A possible solution for the future could be the expansion of the Brain-to-Text functionality, mentioned in [13], for such languages. The most significant task is to create stronger ERP responses to target triggers, they can either be generated with more effective visual stimulus types or better stimulus presentation patterns [31].

3.3 Interstimulus Interval

The effectiveness of the P300-Speller depends highly on the duration of the stimulus is shown in the studies [26], [18]. In [21] the consequences of the changes were analysed.

In [26] and [18] a classic row-column paradigm was used. Whereas, [21] used a random pattern of six flashing characters.

However, in [26] the used matrix for the character selection was not always the same, because also the comparison of different matrix sizes in combination with different interstimulus intervals were investigated. The defined sizes were a 3 x 3 and a 6 x 6 matrix with the 2 different ISI lengths of 175 ms and 350 ms, which created four different conditions.

Additionally, four more flash and dark time combinations were analysed, 31,25 ms/31,25 ms; 31,25 ms/62,5 ms; 31,25 ms/93,75 ms and 62,5 ms/62,5 ms [18]. A 6 x 6 matrix was used with 30 flashes per character and the ratios of flash time to interstimulus interval length were also analysed.

In [21] two experiments were conducted, for the first experiment the ISI times were all different, but flash and dark time were of equal length: 125 ms/125 ms; 62,5 ms/62,5 ms; 31,25 ms/31,25 ms and 15,625 ms/15,625 ms. For the second experiment the flash and dark time were different: 27 ms/27 ms, 82 ms/27 ms, 191 ms/27 ms, 27 ms/82 ms, 27 ms/191 ms. These two experiments were conducted with a 8 by 9 matrix. For the training of the classifier no feedback was given, but for the other sessions the selected character was displayed.

While the results of [18] and [21] showed that a longer ISI time also yielded a higher accuracy of the P300-Speller. The opposite was described in [26], where the accuracy was higher for a ISI of 175 ms compared to 350 ms. The 3 x 3 matrix produced higher accuracy than the 6 x 6 matrix, even when the probability of the target trigger is higher [26].

Moreover, the length of the ISI and dark time had the most significant effect on the accuracy [18]. The dark time increases the delay of the next stimulus and simultaneously the ISI, which makes the target stimulus less frequent [18].

Also, the flash time increased the accuracy of the P300-Speller [18]. Furthermore, the ISI combinations (31,25 ms/31,25 ms; 31,25 ms/62,5 ms and 62,5 ms/62,5 ms) showed no considerable differences, but the 31,25 ms/93,75 ms combination was significantly higher in accuracy. The results presented in [21] for the first experiment had all accuracy values under 80% but the 125 ms/125 ms condition. In contrast, the observed accuracy values for the second experiment are all between 70% and 80%, but the 27 ms/27 ms combination which was under 70%.

The different studies don't show the perfect flash time and dark time combination, it appears to find the optimal one testing different ISI values is necessary [26]. Furthermore, there seems to be a limit for the beneficial behaviour of increasing the ISI timing [18].

Additionally, feedback hidden for the user compared to the feedback shown, did not effect the

performed accuracy [21]. This behaviour would allow the generalisation from offline studies to online studie [21]

Experiments

The following subsections are detailing the approach to test and measured the experimental set-ups for the different tasks which are needed to ensure the reliability of the application results. First some functionality check were done for the interaction between the devices and the application to determine the correctness. One of the parameters evaluated is the sampling rate. Furthermore, the approach to optimize the ISI configurations are detailed as well. The first few sections are focus on the application and devices and the last few sections, beginning with subsection Picture Sets contain the configurations of the parameters for the interstimulus interval and group ranking.

4.1 Sampling Rate

The sampling rate also referred to as sampling frequency is defined as the number of samples in one second. The two devices (*g.USBamp* and *g.Unicorn*) have two different sampling rates, hence, all tests were done twice. All calculations done in the application, are based on the assumption, that the sampling frequency equals the one provided by the BCI devices. Therefore, a thorough analysis of the sampling rate is needed to verify the expected behaviour.

The log file for the EEG signal includes a channel for the time elapsed for each sample, since the start of the software. But these times are the optimal values for the configured sampling rate in the application. For this reason an additional channel with proper time for each sample had to be created. This channel could have been added to the already existing log file, but between the recording of the EEG signal and the saving of it a few signal processing steps are in between. Therefore, a new log file, right after the recording, was created to circumvent possible time changes.

Firstly, the original *Simulink* program was used and an extra block was added to keep the elapsed time start. The added block is a *MATLAB function block* with a user-defined function. Furthermore, another block for saving the elapsed time as a data vector had to be added as well.

Secondly, for each device several different recording time durations were used, which are 1.5 min,

3 min, 5 min, 10 min, 15 min and 30 min. The application has a parameter, that limits the run-time can be limited in combination with the fixed-time between samples leads to a recording with always the same number of samples.

Thirdly, the created data vector was used to compute different parameters. Equation 4.2 shows how the sampling frequency f_s can be calculated either with the mean of the time between samples (T_{mean}) or with the number of samples and the length of the recording in seconds. The subtraction of 1 is needed because the application has at time zero the first sample. Correspondingly, the average time between samples was analysed as well as the elapsed time over the samples.

$$T_{mean} = \frac{\text{duration in seconds}}{\text{number of samples} - 1} \quad (4.1)$$

$$f_s = \frac{1}{T_{mean}} = \frac{\text{number of samples} - 1}{\text{duration in seconds}} \quad (4.2)$$

4.2 Trigger Display

The main part of the functionality of the application involves displaying the different stimuli to the user and how long the the configured settings match the real time runs. In contrast, to section Sampling Rate, the gathering of the required data for the analysis requires much more effort than simply creating a new data vector and saving it within the application. In order to measure the time difference of the stimuli onset in the application and the picture shown to the user an optical sensor was used.

Firstly, a picture set had to be modified, where each image has a white square at the top left corner. For these runs the images themselves are not as important as the fact that an visual stimulus was shown on the display. A normal run shows a picture then a black screen and subsequently another picture and after the configured flash time a black screen again. In fact, this sequence in combination with the modified picture set creates a maximal difference, where the white square is located, between image and black screen. Therefore, the optical sensor was mounted on the screen at exactly the position of the white square of the images. Consequently, the created output of the mounted sensor was a square wave signal.

Secondly, this created signal had to be recorded and saved with the timing of the trigger representation of the application. To achieve this 2 different types for the experimental setup were used, each for one device. For the more advanced device, the *g.USBamp*, the digital input as mentioned in section *g.USBamp* was used. As a result the cap with the electrodes was not needed.

For the other setup, with the *g.Unicorn* device, the recording of the signal was more complicated, because it doesn't have a digital input. Therefore, a self-made connector, mounted on the electrode of a *g.GAMMAcap* was utilised, which made it possible to connect to the output signal of the optical sensor. In both instances the output signal of the sensor was connected to channel one (*Ch1*) of the EEG signal recording devices. In Figure 4.1 the two different setups are depicted. Because the electrode or the digital input voltages are limited, a voltage divider had to be constructed with R_1 and R_2 , which decreased the input voltage for the two different devices to 440 mV, see equation 4.3 and 4.4.

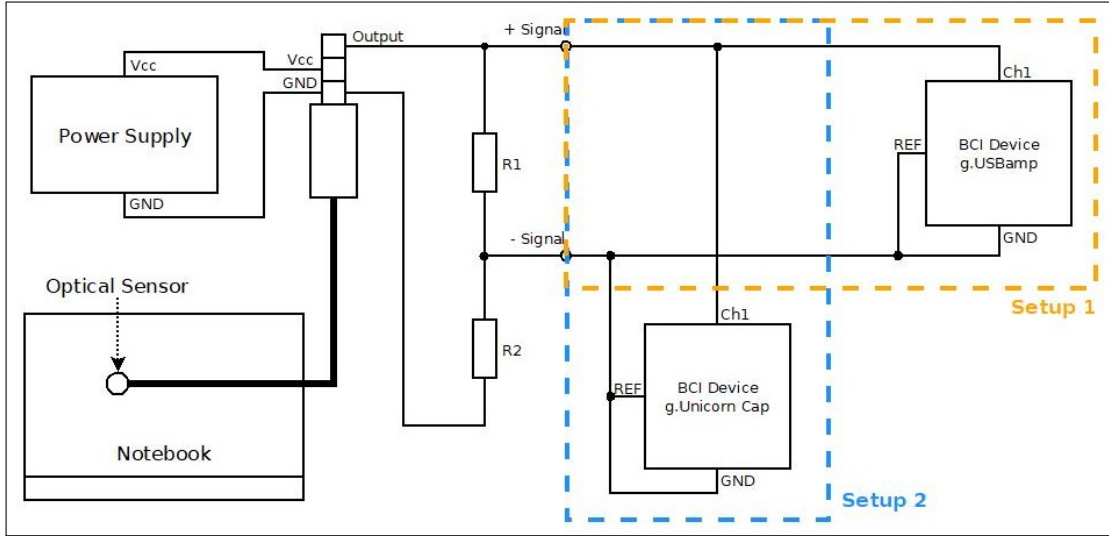


Figure 4.1: The two different setups to measure trigger timing on a notebook screen. *Setup 1* is with the g.USBamp and *Setup 2* uses the g.Unicorn

Thirdly, the filters, notch filter and band-pass filter, had to be removed, to allow the recording for well looking square wave signal. Consequently, the saved signal incorporated the sensor signal, the trigger onset of the application and the timing of each sample. All this information was used to calculate different parameters, such as jitter of trigger, ISI length, flash time and dark time.

$$R_1 = 1 \text{ k}\Omega, \quad R_2 = 4,7 \text{ k}\Omega \quad \text{and} \quad U_{total} = 2,5 \text{ V}$$

$$\frac{U_1}{R_1} = \frac{U_{total}}{R_1 + R_2} \quad (4.3)$$

$$U_1 = \frac{U_{total} \cdot R_1}{R_1 + R_2} = \frac{2,5 \text{ V} \cdot 1000 \Omega}{1000 \Omega + 4700 \Omega} \approx 0,44 \text{ V} \quad (4.4)$$

4.3 Trigger Frame

The trigger frame refers to the samples around a visual stimulus, which begins 100 ms before the trigger and ends 700 ms after the trigger. The task is to determine if the calculations for the LDA classifier and the rankings are as expected.

Moreover, the LDA classifier creation process was used, because it includes all the same processing steps for the ranking as well as the creation of the classifier. The difference between the normally created classifier and the self-programmed one, indicates if the number of samples used are equal and the calculation steps to generate the features are correct. Furthermore, it indicates if the classifier consist of the right bias and weights calculated from the features.

In [7] the process to calculate the LDA classifier parameters is described, textual and with formulas. The following equations (4.5 - 4.11) are either the formula or a modification of one mentioned there.

The first step to calculate the LDA classifier was to split the feature matrix (section *Ranking*, especially equation 4.12) in two parts, one with the features of the target images (D_1) and the other one with the feature of the non-target images (D_2). Then calculate for each feature the mean value, see equation 4.5, where n_i stands for the number of images in each subset.

$$m_1 = \frac{1}{n_1} \sum_{x \in D_1} x, \quad m_2 = \frac{1}{n_2} \sum_{x \in D_2} x \quad (4.5)$$

Secondly, the covariances of both subsets were calculated (equation 4.6) and were added together as S_W (disclosed in equation 4.7).

$$S_i = \sum_{x \in D_i} (x - m_i)(x - m_i)^T \quad (4.6)$$

$$S_W = S_1 + S_2 \quad (4.7)$$

Thirdly, the difference between the two means (equation 4.5) of the subgroups was computed, see equation 4.8.

$$\text{diff} = m_1 - m_2 \quad (4.8)$$

Afterwards, equation 4.9 shows how the different weights (w) for the LDA classifier were calculated.

$$w = S_W^{-1} \text{diff} \quad (4.9)$$

Finally, the last parameter for the classifier was calculated with the equations 4.10 and 4.11, it is either referred to as *bias* or as w_0 .

$$m_{total} = \frac{m_1 + m_2}{2} \quad (4.10)$$

$$w_0 = \text{bias} = \sum -wm_{total} \quad (4.11)$$

Generally speaking, if the calculated weights vector w and the parameter w_0 were identical to the vector and parameter generated via the script provided by *g.tec*, then the signal processing, the number of samples before and after the trigger and the LDA classifier were calculated correctly.

4.4 Set-up

Determination of the effectiveness of these different training sets was done by analysing the accuracy of the created LDA classifiers. For each run the laptop screen was 80 cm away from

the participant and the used configuration for the application was, 150 ms for the flash time as well as the dark time and the number of flashes was set to 30. The application ran in full screen mode on the laptop, which has a display size of 15,6 inch. Additionally, the used BCI device was the g.USBamp, because it is connected to the power grid and can be used for several hours.

4.5 Picture Sets

To analyse the importance of the composition of the picture sets for the training of the LDA classifier different picture sets were created. Therefore, several picture sets had to be made. An essential point to consider for the creation of new picture sets were the copyright laws of the used pictures, because the application will be used commercially in the future. For that reason, different websites exist, where images can be downloaded for free and fall under the *Creative Commons Zero*¹ (CCO) license or similar. More precisely, all images under this license can be used for commercial purposes without permission of or reference to the creator. Additionally, all picture can be modified without interfering with the commercial usage. All new images used for the new picture set were copied from the *Pexels*² website.

In comparison to the originally used picture set for the training of the LDA classifier the new ones use the same 35 grayscale images (Figure 4.4) but the colour image is different for each. Overall, seven new picture sets were created by using different coloured pictures (animal, baby, castle, fire, flower, house and landscape), see Figure 4.2. These images were resized (maximum width of 350 px and a height of 240 px) to fit the application and keep the original picture ratio. Due to layout reasons, the pictures in Figure 4.2 are shown differently sized.



Figure 4.2: The target images for the new picture sets, each image is the target in one new picture set.

Moreover, to eliminate a possible occurring error during the classifier training, each picture set was used twice for a training run. Also, all picture sets were used a second time with the

¹<https://creativecommons.org/>

²<https://www.pexels.com/>

coloured image in grayscale to determine if the coloured image itself, not considering the content of the image, has an impact on the effectivity. Furthermore, for each run the coloured image or the created grayscale image from it, was used as target image.

Additionally, to create reliable ERPs for target and non-target images, a task was given to count the appearance of the target image, this will support the occurrence of a well-formed P300 wave in the EEG signal. Moreover, it was advised to count in the head to eliminate possible disturbances in the EEG signal through muscle movement.

The calculations for creating a classifier as well as generating a Figure to display the accuracy was done by using the *gbsanalyze* tool in combination with an accuracy calculations script, both developed and provided by *g.tec*. Generally speaking the script creates an LDA classifier with an increasing amount of flashes for the training and uses all number of flashes as a test set and calculates the percentage of correctly identified target images.

4.6 Copy Spelling

The application has two different operation modes, one is the *Copy Spelling*, which will be examined in this section and the other is the *Ranking* mode (see next section). As mentioned before the used laptop has the screen size of 15,6 inch and had a distance of 80 cm from the subject. Additionally, the application was executed in full screen on the laptop. Also, the used brain-computer interface device was the *g.USBamp*.

In this section the focus lies on the interstimulus interval and the number of flashes for each picture. Therefore, seven different configurations or rather combinations of flash time and dark time were used with a different number of flashes, table 4.1 illustrates them. Coupled with these combinations different number of flashes were applied for each. More precisely, every configuration was recorded with the following amount of flashes: 3, 5, 7, 10, 12, 15 and 20.

Configuration	1	2	3	4	5	6	7
Flash Time [ms]	50	50	75	75	75	100	150
Dark Time [ms]	50	75	50	75	100	75	150

Table 4.1: The different ISI configurations used.

At first, a new LDA classifier had to be created for each new flash and dark time configuration. Hence, a picture set was used with 1 coloured image of a baby (see Figure 4.3) and 35 grayscale images, which are shown in Figure 4.4.

The coloured image was the target for the training. A training run was conducted with the selected ISI configuration and with 30 flashes. Then, the resulting ERPs, for the target and non-target were looked at to determine if the created ERPs would create a good classifier. If these ERPs were not well-formed, P300 was not identifiable, another training run was conducted. However, were these ERPs well-formed, a script, provided by *g.tec*, was used to calculate the LDA classifier from the recorded log file. Additionally, the created classifier was tested by using the same picture set including the coloured image as a target for five times. This extra test was conducted with only three flashes. If the result was 100% then the classifier was used.



Figure 4.3: Target image for classifier creation.

After the creation of the classifier a different picture set was used, which consisted of 36 coloured images, see Figure 4.5.

These images were downloaded from the *Pexels* website and then resized to keep the ratio and fit the size used in the application. For every run the same five target images were selected, see Figure 4.6.

For every target change during a run, the application shows a black screen for three seconds and then the target image for another three seconds before the run continues.

At the end of each run the application shows how many of the targets were correctly identified as well as a percentage value of the identification, for the whole run. These results were saved for every run. Furthermore, for each configuration and setting for the different number of flashes five test run were carried out.

4.7 Ranking

In this section the ranking of the images in one picture set will be described. As mentioned in the sections *Picture Sets* and *Copy Spelling* the first step was to create an LDA classifier. This was done by using a picture set of 1 coloured image and 35 grayscale images. The target image was a picture of a baby (Figure 4.3) and the grayscale images are depicted in Figure 4.4. Again, 30 flashes were used for the classification run and the coloured image was the target image. A

classifier was only created if the ERPs of the training runs showed the needed characteristics of the P300.

In the next step two new picture sets were created, all images were downloaded with the CC0 license from the *Pexels* website. Moreover, each picture set consists of 31 grayscale images and 5 coloured ones, images were resized to fit the maximum dimensions of the application with a width of 350 px and a height of 240 px.

For the first picture set, the coloured images picked were all about food (Figure 4.7) and were used in seven different flash and dark time configurations, see table 4.1.

Each run of one of these configurations was done with 20 flashes per image and repeated five times. The remaining images of the picture set are depicted in Figure 4.8. Additionally, in this mode the task is to concentrate on every image with the same amount of intensity. After the *food* picture set was done the second one was used, which has five coloured pictures of babies (see Figure 4.9), and replace the coloured images of food in the first set.

Afterwards all recorded log files in combination with the LDA classifier were used to calculate the ranking for the different number of flashes, from 1 to 20 per run. This process was done by iterating over the number of flashes for the responses to the images.

For every number of flashes a feature matrix was created, see equation 4.12. In addition the equation 4.13 shows how the score value for each image was calculated. These *Score* values were used for the ranking, from highest to lowest. For both equations n stands for the number of features, m is number of different images, w_0 refers to the bias of the LDA classifier and w_1 to w_n are the different weights for each feature. With this process, rankings for the images were calculated for the different number of flashes. Also, the generated rankings for the different number of flashes (1 to 20) were saved in a text file which corresponds the resulting text file created by the application in the ranking mode.

Furthermore, the top 5 images for each picture set (*food* and *babies*) were plotted for the same 7 different number of flashes as the ones used in section Copy Spelling: 3, 5, 7, 10, 12, 15 and 20. For generating the collective ranking for each amount of flashes the median and mean vales of the ranking positions were used.

$$FM_{m,n} = \begin{pmatrix} f_{1,1} & f_{1,2} & \cdots & f_{1,n} \\ f_{2,1} & f_{2,2} & \cdots & f_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ f_{m,1} & f_{m,2} & \cdots & f_{m,n} \end{pmatrix} \quad (4.12)$$

$$Score_m = \sum_{i=1}^n (w_i \cdot f_{m,i}) + w_0 \quad (4.13)$$

For the calculations of the weights (w) and w_0 see section *Trigger Frame* and equations 4.5 - 4.11.

4.8 Group Ranking

For the group ranking the results of the ranking mode of the application with different subjects were used. After the BCI head cap was put on, every subject had to do the copy spelling mode

to create the LDA classifier. As usual the picture of the baby (Figure 4.3) was used as the target and 35 grayscale images (Figure 4.4) were the non-target pictures. The configuration for both the copy spelling and ranking mode was 100 ms for the flash time and 75 ms for the dark time. Moreover, after the classifier was created the application was started again and the coloured picture of the baby had to be identified five times to make sure the classifier was good enough. Otherwise the whole process for the creation of the classifier had to be done again. Additionally, for the LDA classifier creation 20 flashes and for the ranking 10 flashes were used and the distance from the laptop screen was 80 cm.

Furthermore, two different picture sets were used for the ranking, the first one is the *food* picture set (see Figure 4.7) and the second one the *baby* picture set (see Figure 4.9). Both of these two coloured image sets were used in combination with 31 grayscale images (Figure 4.8) to create two picture sets with a total of 36 images. For each picture set five runs were carried out and all results were saved. The result of one ranking consists of an application window with the trigger images and their ranking position, which was saved via a screenshot as a picture and a text file with the ranking position and the trigger image file name.

These resulting text files were used to calculate an overall rank for each image. In fact, two different methods were used to calculate the group ranking. Firstly, the *mean* value were all ranking positions of one trigger were added and then divided by the number of rankings. Secondly, the *median* value were all ranking positions are sorted from lowest to highest value and the value, which is positioned in the middle of the series, was selected. If there are two values in the middle of the series the mean values of those two was calculated.

Besides, the calculated ranking result for 10 flashes per image, additional rankings with 3, 5, and 7 number of flashes for each images were also done. For a better presentation of the results only the top 5 images for each number of flashes setting were shown in the resulting MATLAB window.

One Subject Results

After the ranking runs for one subject were done the group ranking was created with only the subject's results for both picture sets with different number of flashes. As already mentioned 2 different parameter *mean* and *median* were used as well as different number of flashes.

Three Subjects Results

An overall ranking was created with the results of all the subjects, which also shows the top 5 images for the 4 different number of flashes (3, 5, 7 and 10). In addition to the two already used parameter (mean and median) two extra parameters *mean absolute deviation (MAD1)* and *median absolute deviation (MAD2)* were used. To determine the ranking for images, with the same rank the additional parameters were applied to make it possible to rank these images as well. These extra parameters were used in combination with the mean and median parameter, which created 4 different ranking: mean, median, mean + MAD1 and median + MAD2. Again, two different picture sets with 5 coloured images and 31 grayscale images were used.

Furthermore, a summary of the different ranking parameter was created, but it only shows the result for 10 flashes per image.

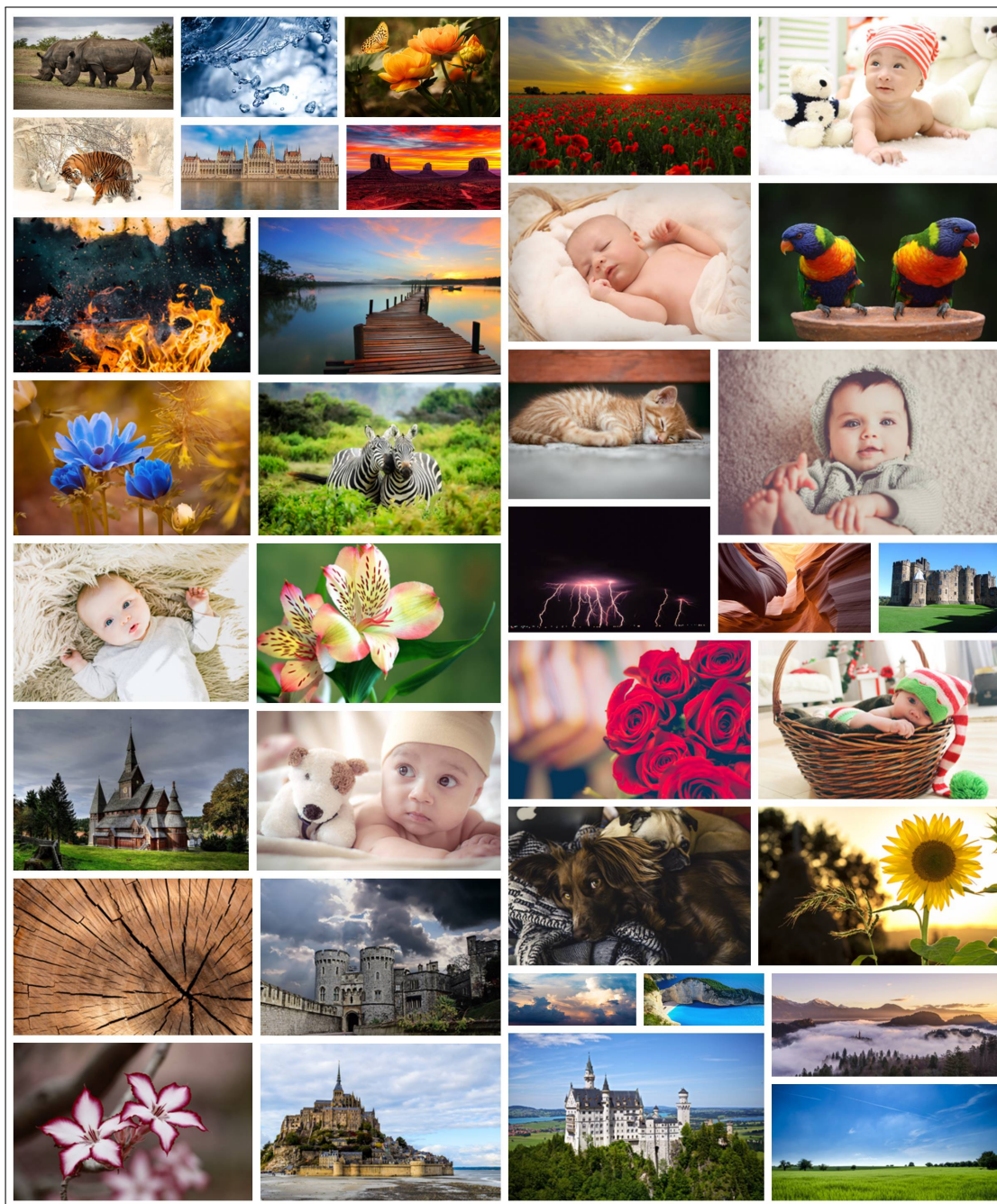


Figure 4.5: The picture set used in the *Copy Spelling* mode for ISI analysis.



Figure 4.6: The five target pictures used for each run in the *Copy Spelling* mode.



Figure 4.7: The five coloured images of the *food* picture set for the ranking mode



Figure 4.8: The remaining images of the picture set for the *Ranking* mode



Figure 4.9: The five coloured images of the *baby* picture set for the ranking mode

Results

In this section the results of the experiments described in the previous section are shown and in the following section these results will be interpreted.

5.1 Sampling Rate

The recorded data from the g.Unicorn device, test duration length of 5 min, were used to show the elapsed time over all samples, see Figure 5.1. In this Figure the sampling frequency is faster for the first few seconds and decreases to a lower frequency, which seems to be similar to the optimal frequency.

Figure 5.2 depicts the first 2500 samples to highlight this odd behaviour at the beginning of the data.

In comparison, to the g.Unicorn, the data of g.USBamp shows similar behaviour for the first few seconds. The recorded data for the g.USBamp is 5 min long, see Figures 5.3 and 5.4.

Furthermore, the time difference of each sample between the optimal and the measured time values is displayed in Figure 5.5 for the g.Unicorn and in Figure 5.6 for the g.USBamp. Both data sets have the length of five minutes and consist of two plots. The upper plots of both Figures (5.5 and 5.6) show the time difference from the first to the last sample. In contrast, the lower plot for each Figure (5.5 and 5.6) shows the time difference from the 1501st to the last sample. This cutoff eliminates the first few seconds of the data, which show a different behaviour. In the Figures 5.2 and 5.4 the start sample, for the cutoff, are depicted with a dashed vertical line.

Additionally, the times between the measured samples are also analysed, in Figure 5.7 the left plot shows the time differences between each sample for all samples and the right plot the time differences between each sample for sample number 10000 to sample number 10250. As usual, the duration length of the measured data is 5 minutes and the devices used was the g.Unicorn.

The same settings are used in Figure 5.8, the only difference is that the data were recorded with the g.USBamp. The mean time between two samples is for all recorded data lengths always 4 ms for the g.Unicorn and 3,9 ms for the g.USBamp.

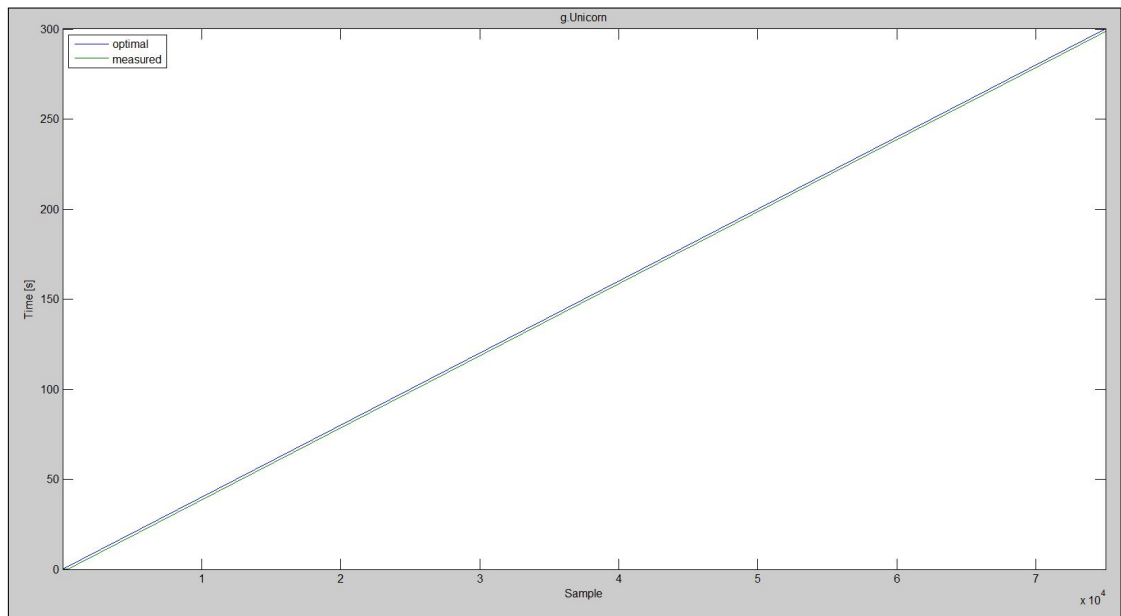


Figure 5.1: Time elapsed for all samples for the g.Unicorn and a duration length of 5 min. The blue and green curve represent the *optimal* and *measured* values, respectively. It shows the time elapsed for each sample. (Note: X-axis has a multiplication factor of 10000.)

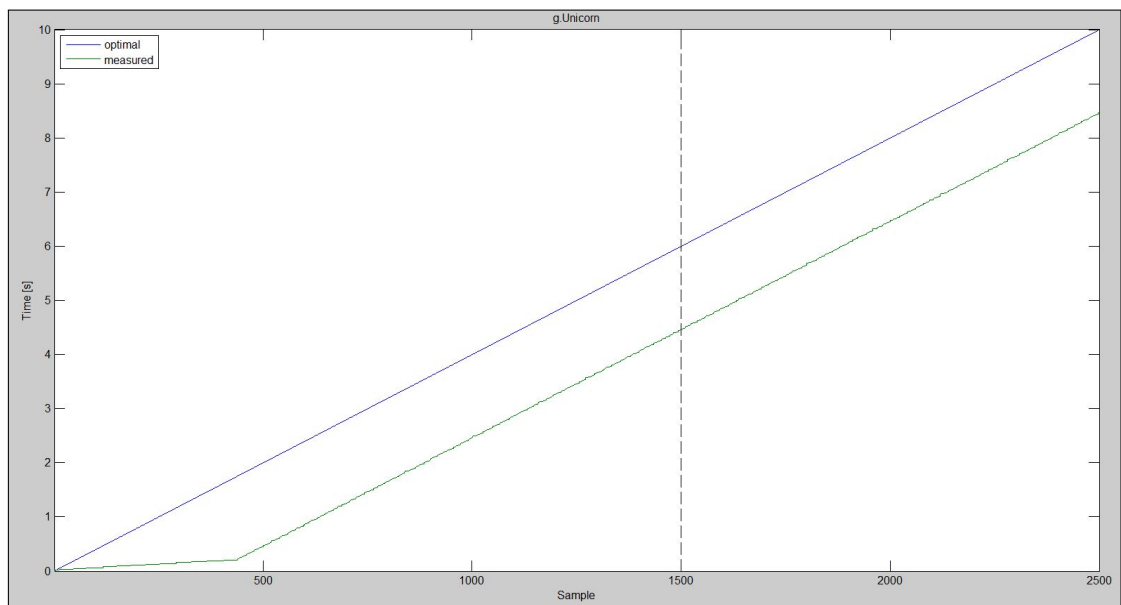


Figure 5.2: Time elapsed for the first 2500 samples for the g.Unicorn and a duration length of 5 min. The blue and green curve represent the *optimal* and *measured* values, respectively. It shows the time elapsed for each sample for the first 2500 samples.

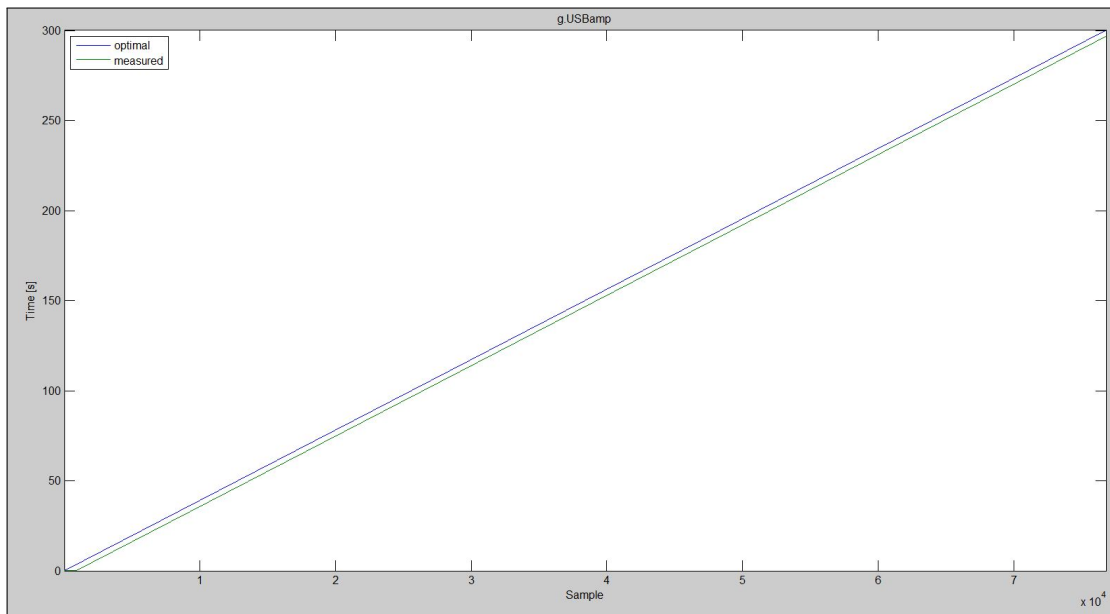


Figure 5.3: Time elapsed for all samples for the g.USBamp and a duration length of 5 min. The blue and green curve represent the *optimal* and *measured* values, respectively. It shows the time elapsed for each sample. (Note: X-axis has a multiplication factor of 10000.)

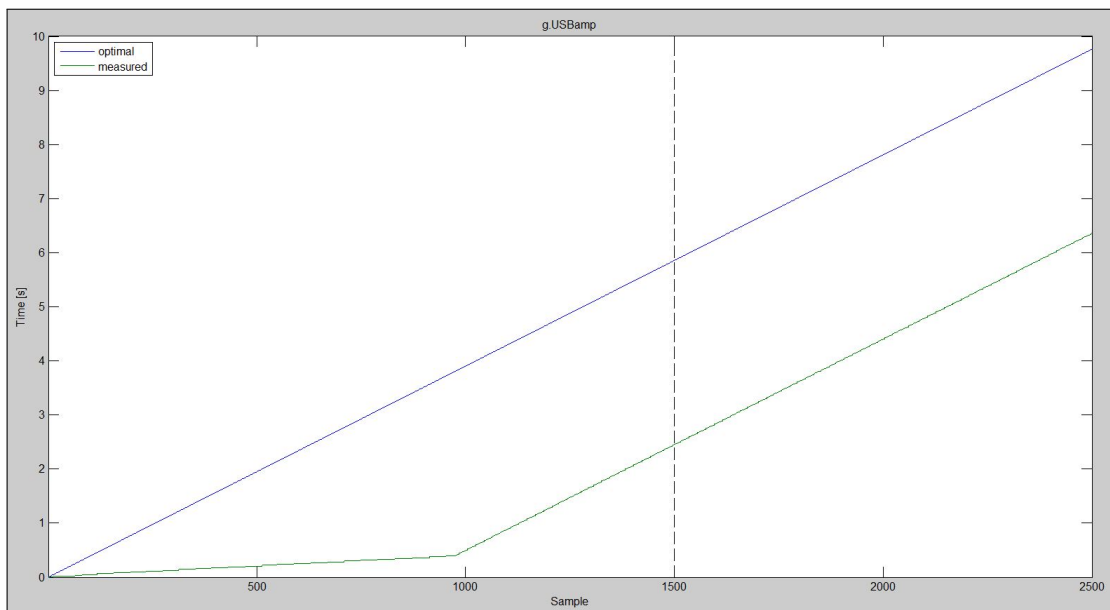


Figure 5.4: Time elapsed for the first 2500 samples for the g.USBamp and a duration length of 5 min. The blue and green curve represent the *optimal* and *measured* values, respectively. It shows the time elapsed for each sample for the first 2500 samples.

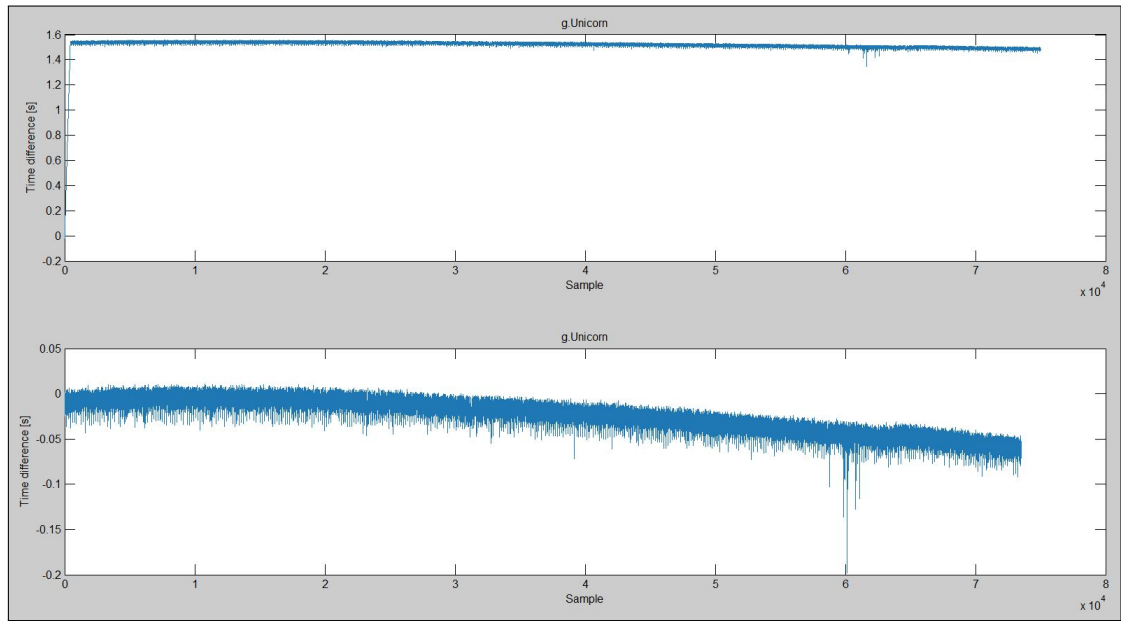


Figure 5.5: Time difference for each sample between optimal and measured time values for the g.Unicorn and a length of 5 min. The upper plot shows all samples from first to last and the lower one from the 1501st to the last. (Note: X-axis has a multiplication factor of 10000.)

Moreover, the normal distribution of the time difference between optimal and measured time values for each sample were calculated and are depicted in Figure 5.9, for both the g.Unicorn and the g.USBamp. For both the devices the length of the recording was 5 min.

Also, a normal distribution of the same data recording was created, where the first 1500 values were cutoff. This first 1500 values include the peculiar behaviour for the first few seconds. The result of this shortened data sets is shown in Figure 5.10.

The equation 4.2 was used to calculate the frequencies of the measured data sets. Several different duration lengths were used to analyse the sampling rate. Because of the odd behaviour of the first few seconds two different sampling rates were calculated. In table 5.1 the frequencies for the all data set samples as well as the shorted data set (cutoff of the first 1500 samples) listed for both the g.Unicorn as well as the g.USBamp.

5.2 Trigger Display

The recorded data sets for the g.Unicorn and g.USBamp were used to calculate the time difference between onset of trigger in the application and the presentation of this trigger on the computer display. The resulting normal distributions of this time delay for both devices is depicted in Figure 5.11. For the data set in this Figure, the flash time and dark time were set to 150 ms.

Furthermore, the different lengths of the time interval between two triggers on the computer screen were analysed for the g.Unicorn and the g.USBamp, see Figure 5.12. Because the time

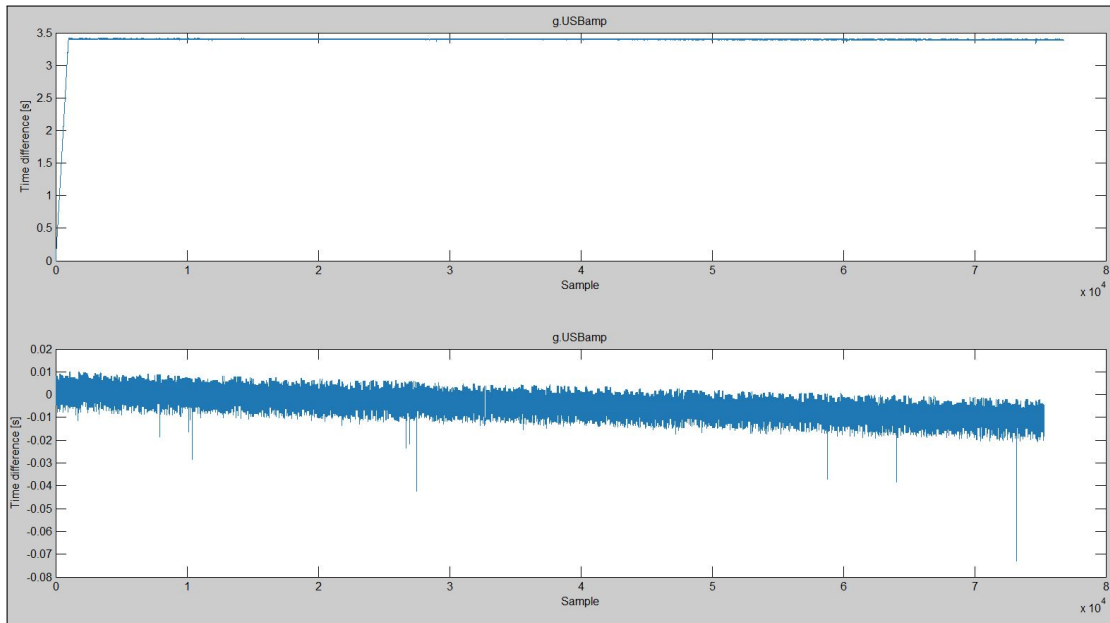


Figure 5.6: Time difference for each sample between optimal and measured time values for the g.USBamp and a length of 5 min. The upper plot shows all samples from first to last and the lower one from the 1501st to the last. (Note: X-axis has a multiplication factor of 10000.)

Duration	Coniguration 1		Configuration 2	
	total	truncated	total	truncated
1,5 min	254,4126 Hz	249,7532 Hz	266,1338 Hz	255,9958 Hz
3 min	251,7961 Hz	249,7345 Hz	260,2140 Hz	256,0112 Hz
5 min	251,2361 Hz	249,9396 Hz	258,9335 Hz	255,9962 Hz
10 min	250,2749 Hz	249,7452 Hz	257,4779 Hz	255,9886 Hz
15 min	250,2589 Hz	249,8379 Hz	256,8619 Hz	255,9904 Hz
30 min	250,0128 Hz	249,7938 Hz	256,5747 Hz	255,9900 Hz
Average	251,3319 Hz	249,8007 Hz	259,3660 Hz	255,9954 Hz

Table 5.1: The calculated sample rates for both configuration 1 and 2, with different duration lengths, which use the two devices g.Unicorn and g.USBamp, respectively .

interval between two triggers for the g.USBamp are all the same, the resulting curve, normal distribution, does not exist. The mean value is depicted in Figure 5.12 at 312,5 ms with a vertical dashed line. Additionally, the duration of the trigger presentation, flash time, on the monitor screen was set to 150 ms, see Figure 5.13. Lastly, the duration in which the screen is black, which is also called dark time, was measured with a set dark time of 150 ms. Figure 5.14 shows this measured duration for both devices.

All the calculated mean values of the different time intervals are shown in table 5.2 and table 5.3 for the g.Unicorn and the g.USBamp, respectively. They also list the times for all the

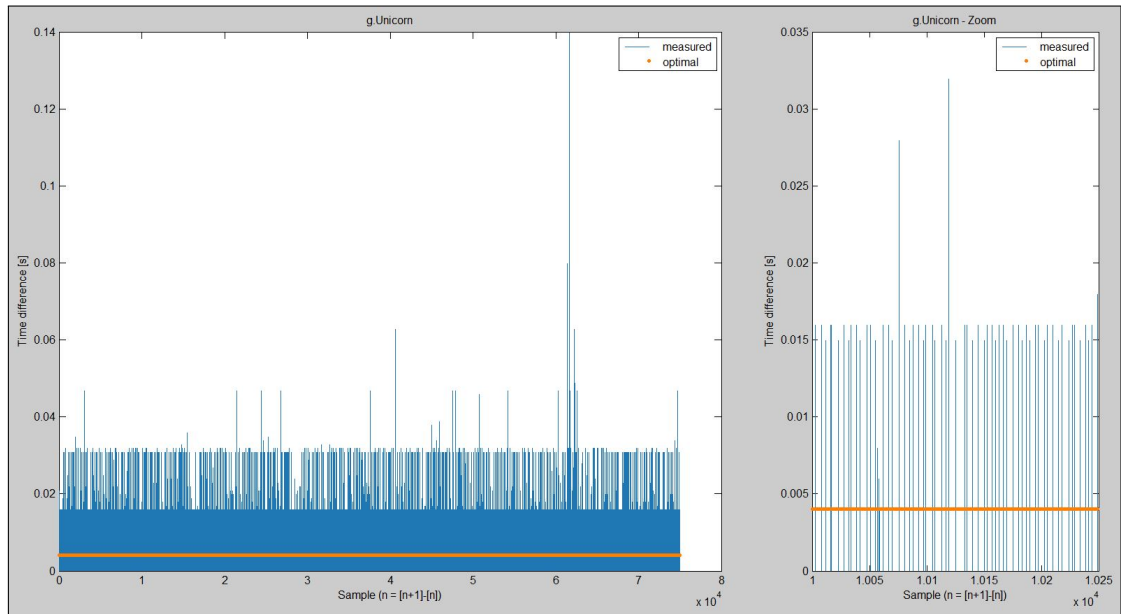


Figure 5.7: Time difference between each sample of the measured data of the g.Unicorn with a duration length of 5 minutes. The left plot shows the time differences between each sample for all samples and the right plot the time differences between each sample for sample number 10000 to sample number 10250. (Note: X-axis has a multiplication factor of 10000.)

other flash and dark time configurations set in the application. The *Jitter of Trigger* refers to the time elapsed between trigger onset in the application and the presentation of the trigger on the computer screen. For the *Trigger to Trigger*, *Flash Time* and *Dark Time* the time deviation to the configured one in the application is displayed.

Deviation [ms]	Flash Time [ms] / Dark Time [ms]					Average [ms]
	50/50	75/75	100/100	125/125	150/150	
Jitter of Trigger	57	57,7	58	58,3	58,5	57,9
Trigger to Trigger	26,8	3,2	22,6	4,5	19,4	15,3
Flash Time	-6,3	-16	-8,5	-17,6	-10,6	-11,8
Dark Time	36,9	19,9	31,1	22,1	30	28

Table 5.2: The measured time intervals for the g.Unicorn with different configurations. The values of last three rows in the table are relative to their configuration times.

5.3 Trigger Frame

The results of three different copy spelling runs were used to calculate for every run a weight vector and a bias. For each run two weights and their corresponding bias was calculated, one

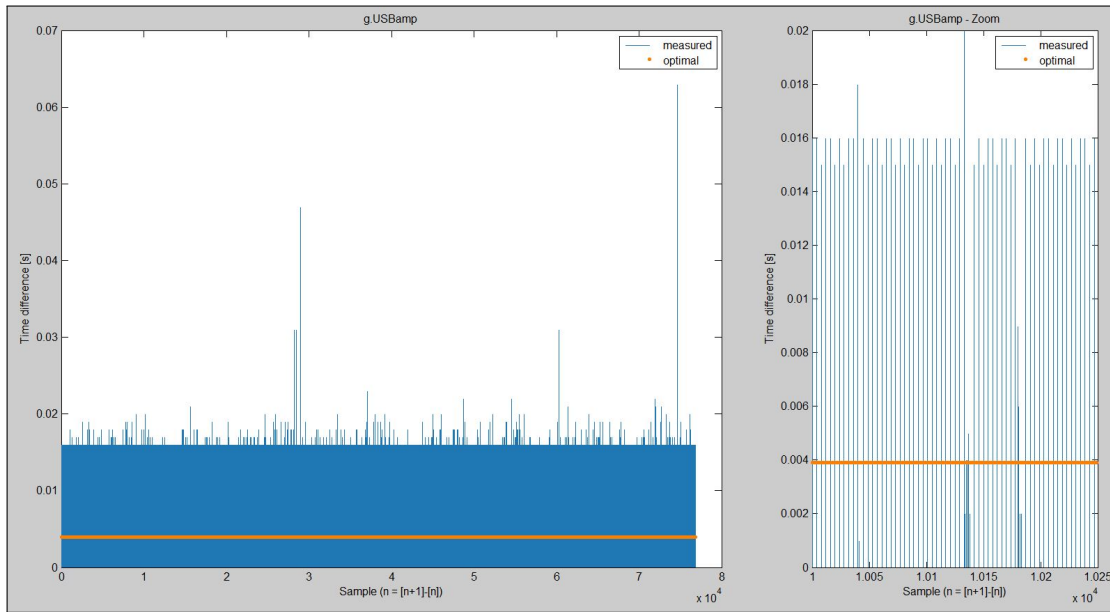


Figure 5.8: Time difference between each sample of the measured data of the g.USBamp with a duration length of 5 minutes. The left plot shows the time differences between each sample for all samples and the right plot the time differences between each sample for sample number 10000 to sample number 10250. (Note: X-axis has a multiplication factor of 10000.)

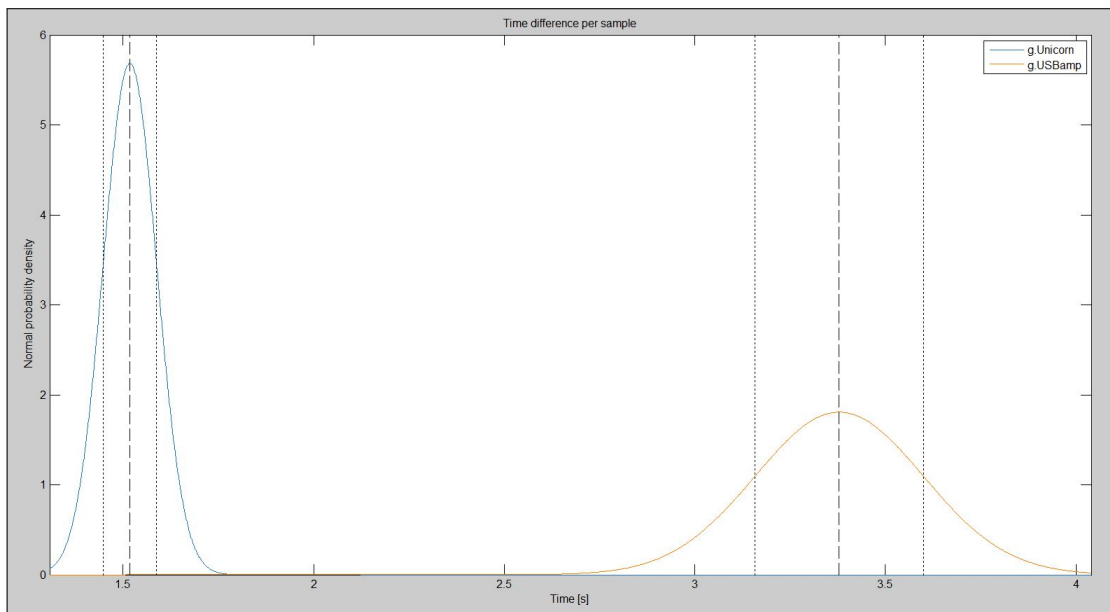


Figure 5.9: Time difference between optimal and measured time values for each sample.

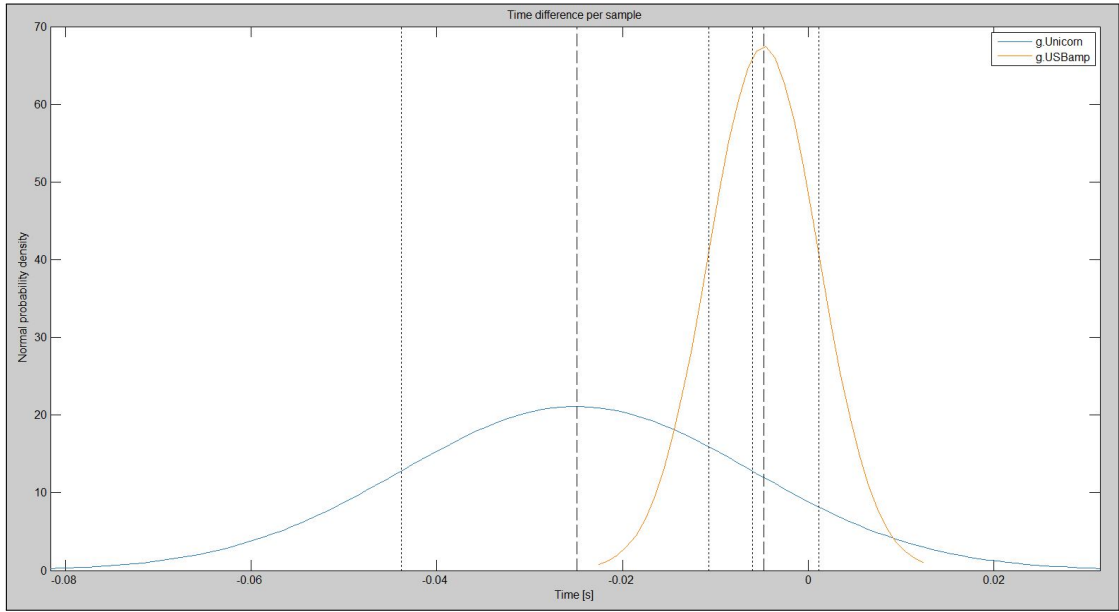


Figure 5.10: Time difference between optimal and measured time values for each sample beginning with the 1501st value.

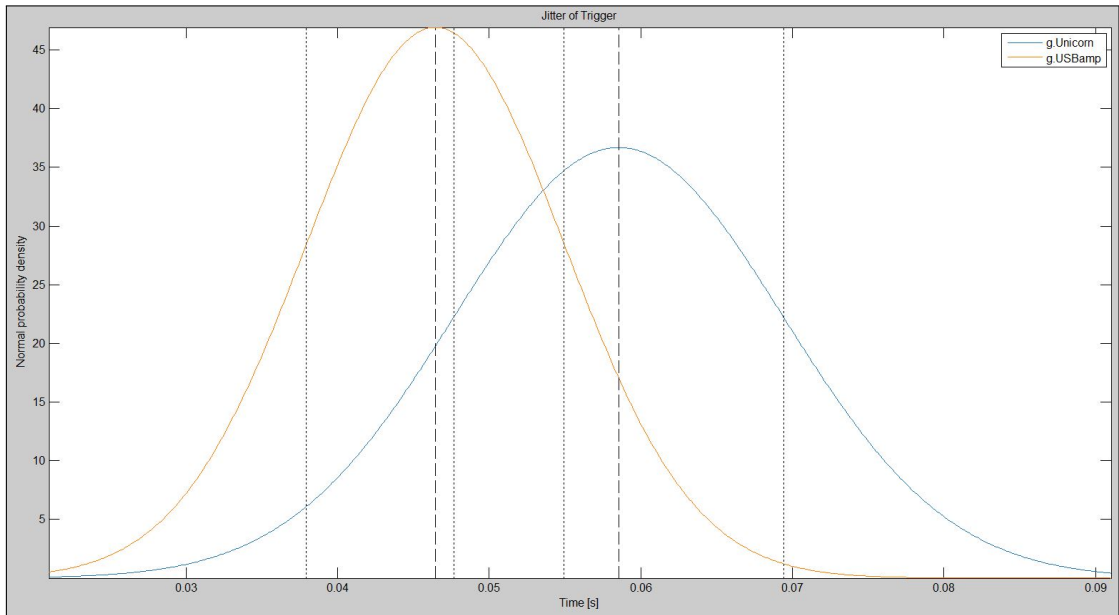


Figure 5.11: Time delay between trigger onset in application and presentation on computer display. The configuration for both flash time and dark time was set to 150 ms.

with the provided script of *g.tec* and one version with the self created methods (section *Trigger Frame*), the results are the same and illustrated in Figure 5.15. The figures shows the two weights

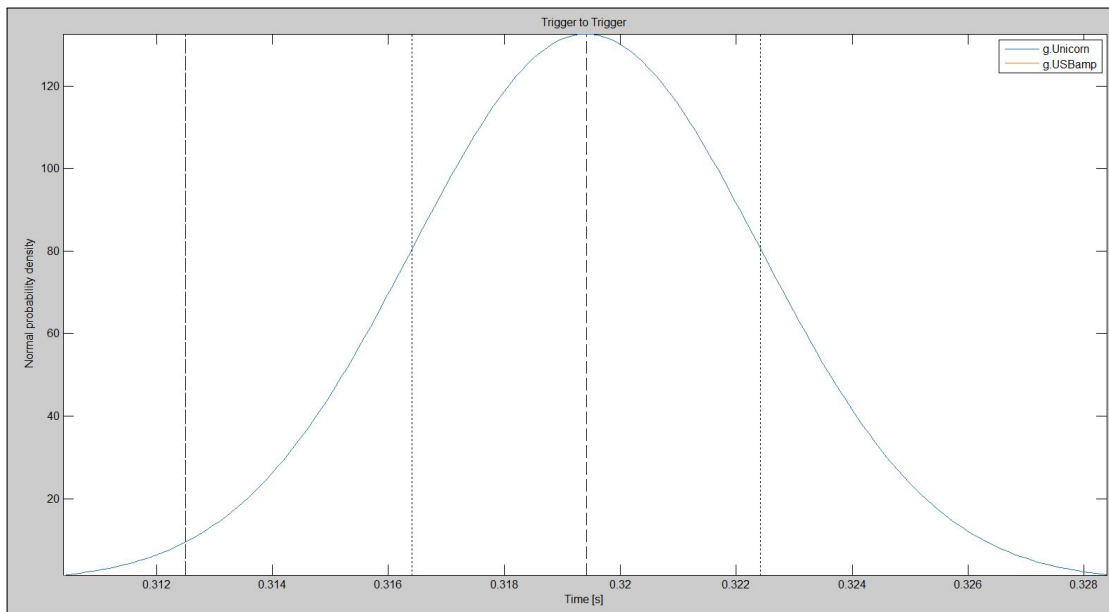


Figure 5.12: Elapsed time between two triggers measured on the computer display. The configuration for both flash time and dark time was set to 150 ms. The g.USBamp values are all the same, therefore no normal distribution is generated. Only the dashed vertical line at 312,5 ms.

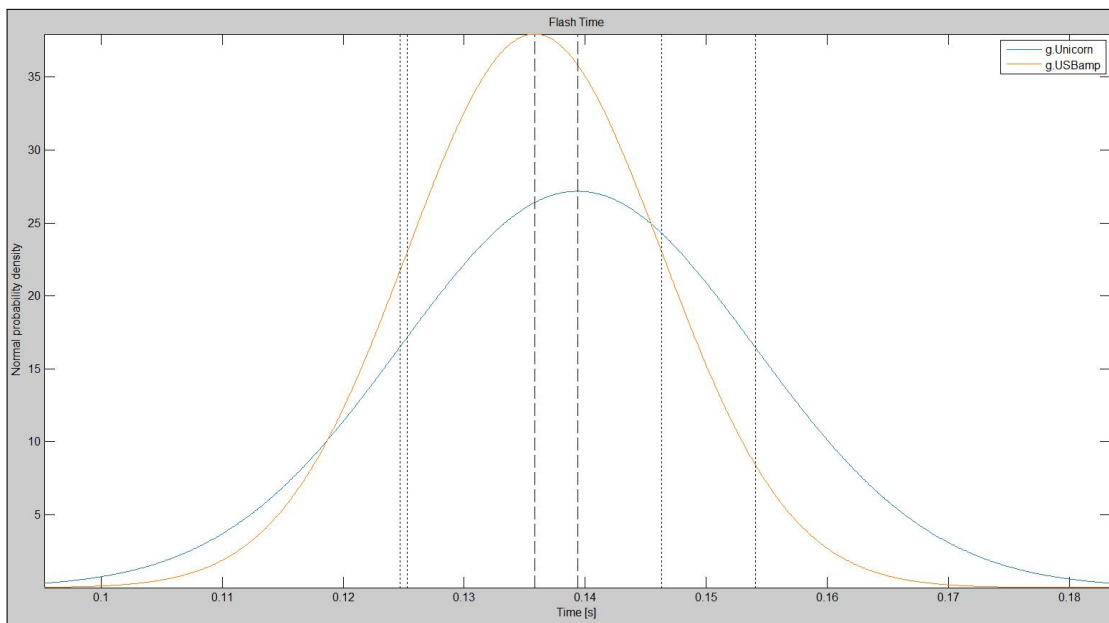


Figure 5.13: Length of flash time on computer screen.

overlapping each other and the bias values are shown over the curve.

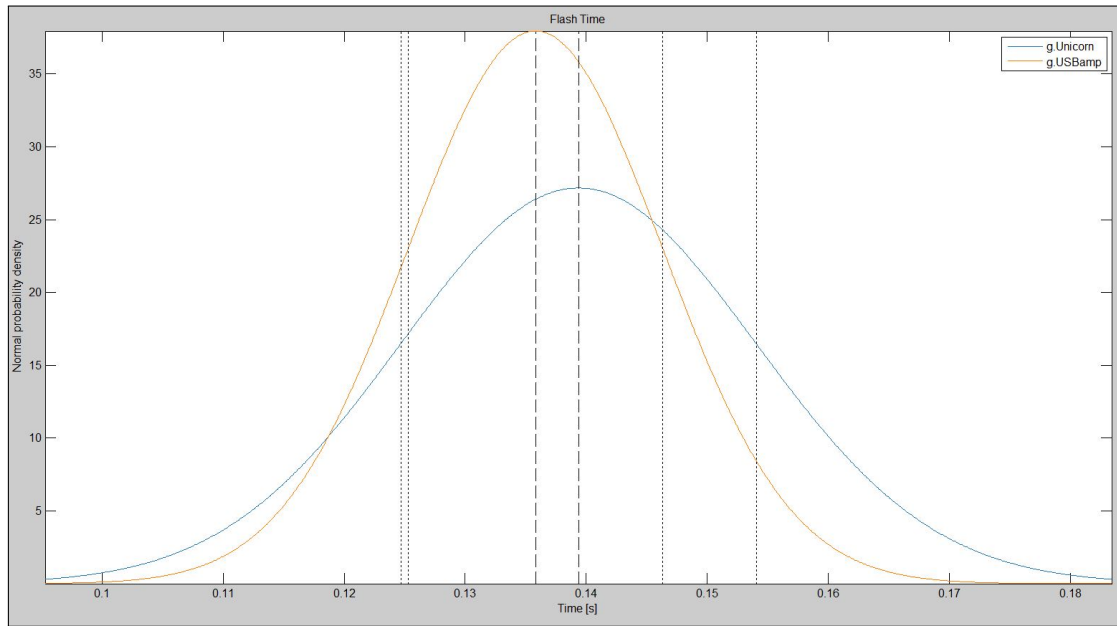


Figure 5.14: Dark time measured on the computer screen.

Deviation [ms]	Flash Time [ms] / Dark Time [ms]					Average [ms]
	50/50	75/75	100/100	125/125	150/150	
Jitter of Trigger	46,6	45,8	46,1	49,9	46,4	46,96
Trigger to Trigger	25	6,2	18,8	31,2	12,5	18,74
Flash Time	-8,7	-16,2	-12,2	-9,2	-14,2	-12,1
Dark Time	33,7	22,4	31	40,4	26,7	30,84

Table 5.3: The measured time intervals for the g.USBamp with different configurations. The values of last three rows in the table are relative to their configuration times..

5.4 Picture Set

The resulting accuracy of both runs of the coloured target image (baby) are show in Figures 5.16 and 5.17. The flash time as well as the dark time were 150 ms and the number of flashes used was 30.

In table 5.4 all the results of the runs are listed, the number indicates how many flashes are needed to get an accuracy of 100 %.

5.5 Copy Spelling

Figure 5.18 depicts a 3D bar plot, which shows the accuracy depending on the configuration of the flash time, dark time as well as number of flashes. The accuracy is generated by the mean

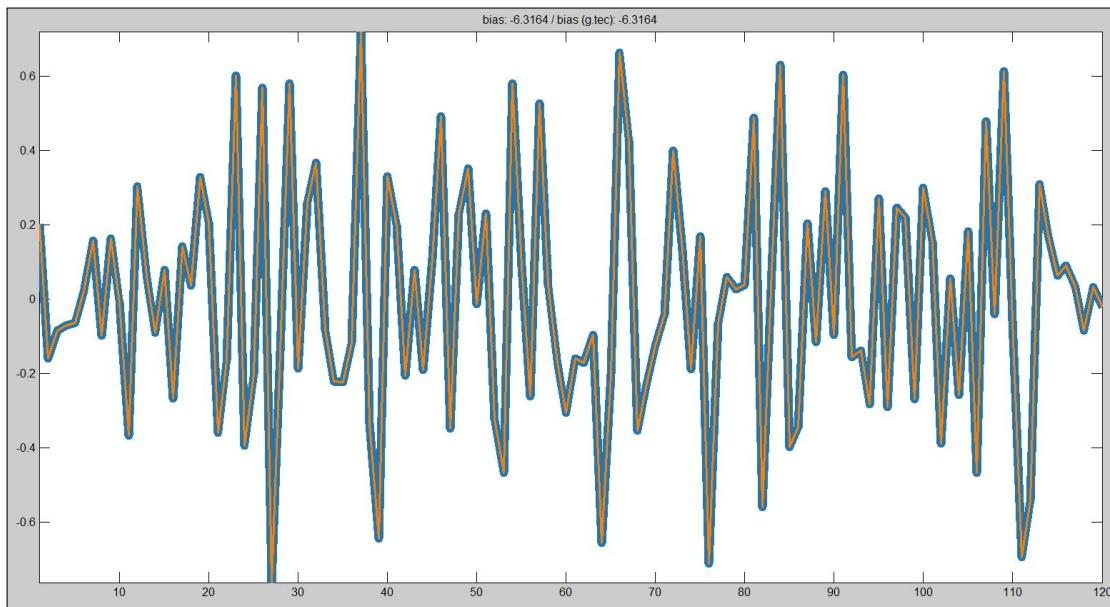


Figure 5.15: Results of the LDA classifier parameters (weights and bias) with the two different approaches for one copy spelling run. X-axis shows the number of feature and the y-axis the weights.

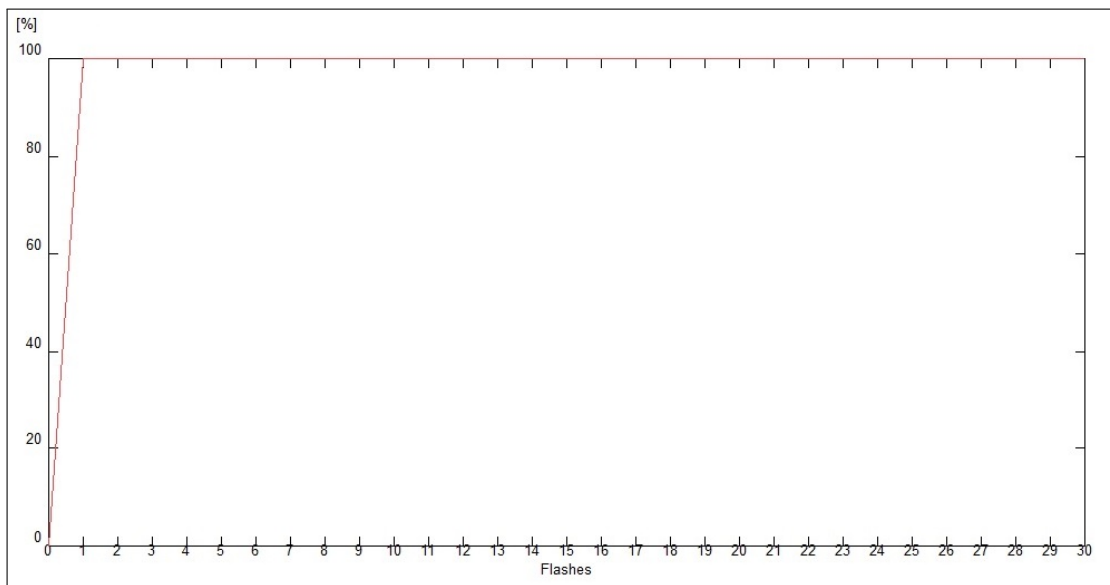


Figure 5.16: Accuracy for the coloured baby target image, first run.

value of all the copy spelling mode results with the same configuration. In table 5.5 are all values listed, which are depicted in Figure 5.18.

Additionally, the number of times a certain target images was correctly identified is illus-

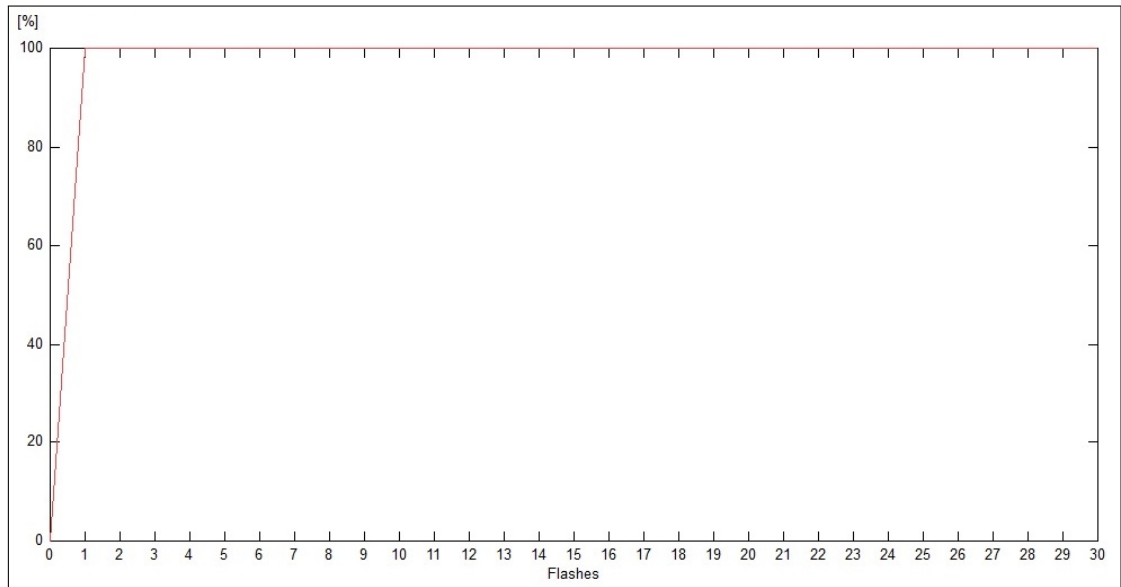


Figure 5.17: Accuracy for the coloured baby target image, second run.

Target	Colour 1	Colour 2	B/W 1	B/W 2	Average
Animal	1	1	1	1	1
Baby	1	1	1	1	1
Castle	1	1	1	1	1
Fire	1	1	1	1	1
Flower	1	1	1	1	1
House	1	1	1	1	1
Landscape	1	1	1	1	1

Table 5.4: Number of flashes needed to generate an accuracy of 100 % for the picture sets with different target images.

trated in Figure 5.19.

5.6 Ranking

The Figure 5.20 shows the top 5 images for the food picture set and a configuration of 150 ms for the flash time as well as the dark time with the mean as ranking parameter. In contrast, Figure 5.21 depicts the top 5 results with the median as ranking parameter.

Likewise, the top 5 images for the configuration of 150 ms for both the flash time and dark time is illustrated for the baby picture set and the mean parameter, see Figure 5.22. The same picture set and configurations are shown in Figure 5.23 but with the median as ranking parameter.

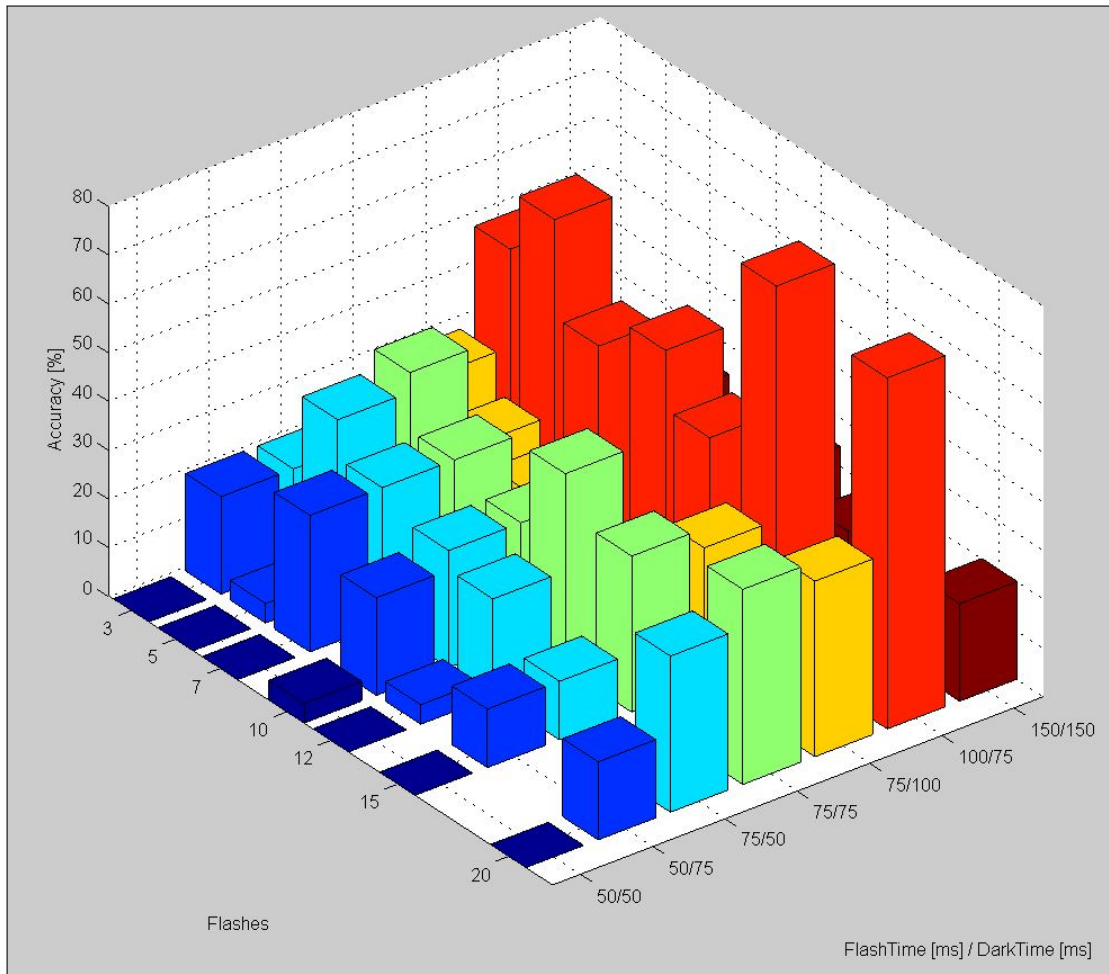


Figure 5.18: Accuracy for the different configurations of the copy spelling settings.

FT [ms] / DT [ms]	Number of Flashes						
	3	5	7	10	12	15	20
50/50	0	0	0	4	0	0	0
50/75	20	4	28	20	4	12	16
75/50	20	36	28	24	20	12	32
75/75	20	40	28	24	40	32	40
75/100	28	20	16	8	12	28	36
100/75	48	60	40	48	36	76	72
150/150	20	24	24	4	24	20	20

Table 5.5: Accuracy table with all values in percentage for the different configurations. *FT* and *DT* stand for flash time and dark time, respectively.

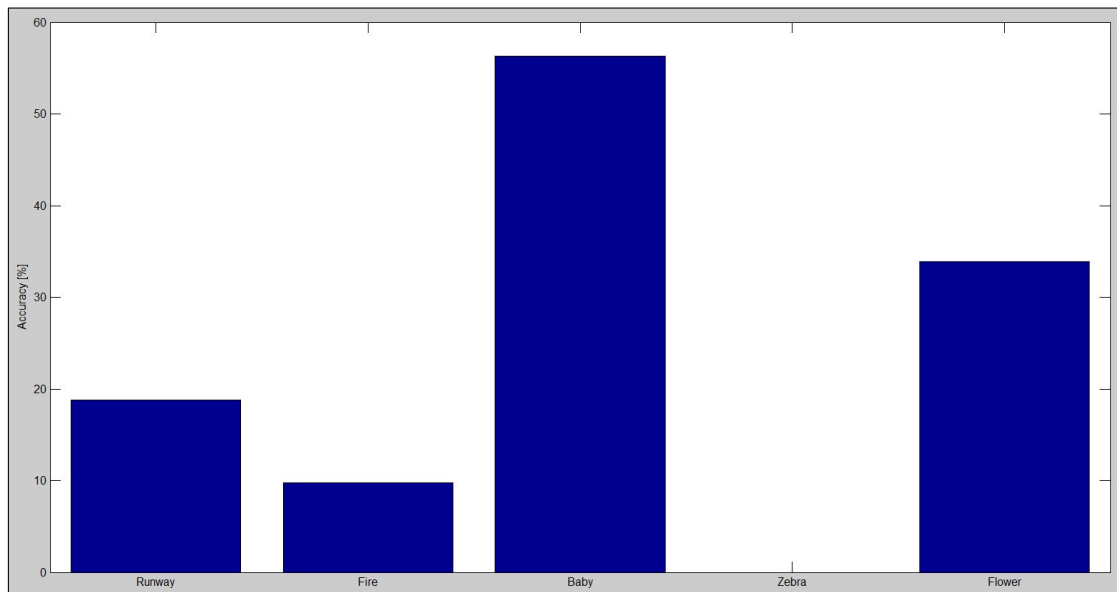


Figure 5.19: Identification of the different target images in the copy spelling mode.

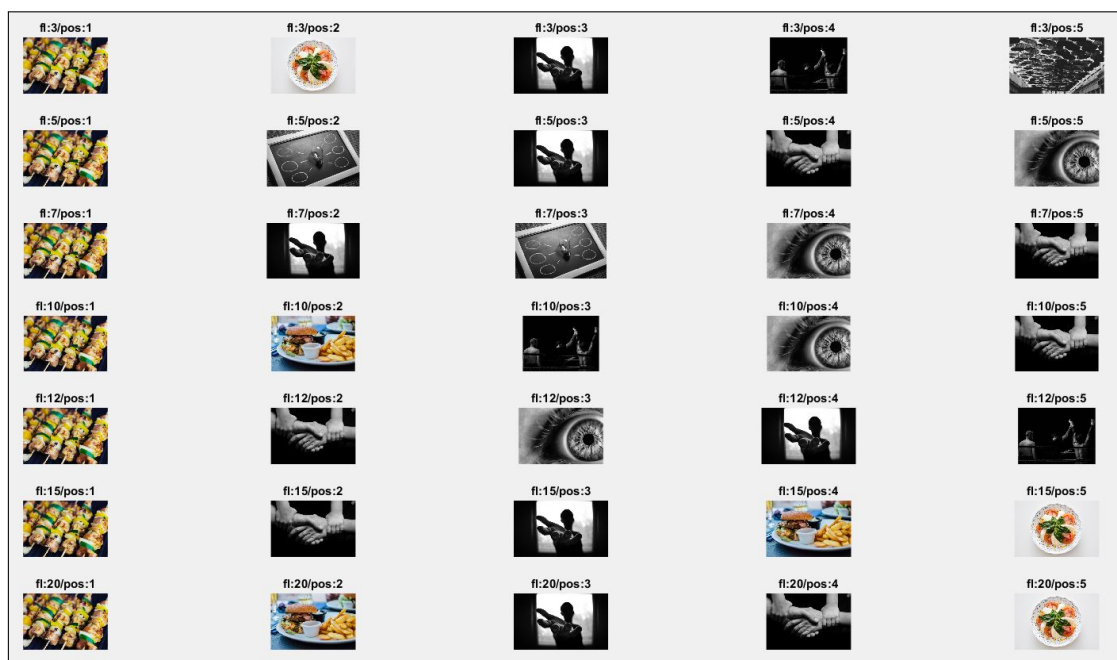


Figure 5.20: Top 5 images of the food picture set with different numbers of flashes and a flash time and dark time of 150 ms with the mean as the ranking parameter.

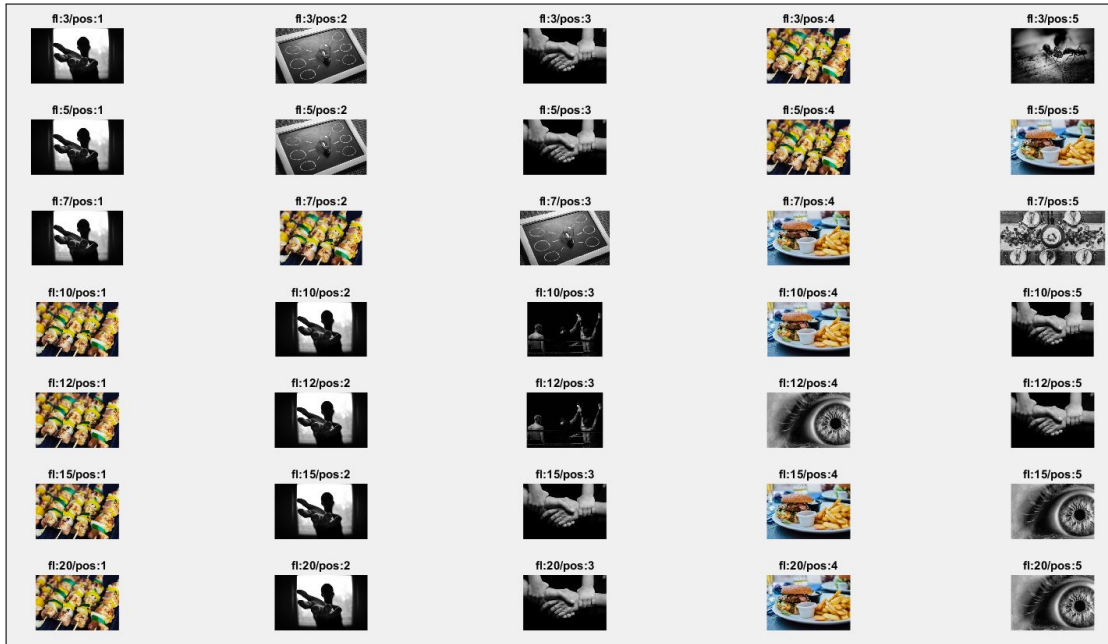


Figure 5.21: Top 5 images of the food picture set with different numbers of flashes and a flash time and dark time of 150 ms with the median as the ranking parameter.

5.7 Group Ranking

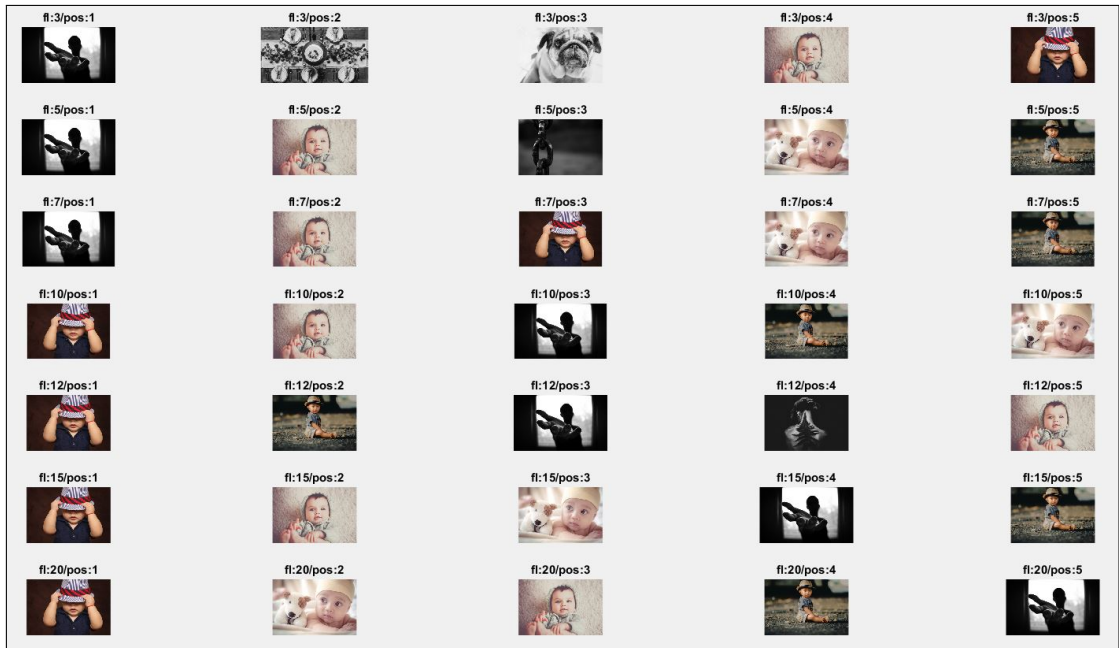
One Subject

Both picture sets *food* and *baby* are illustrated in Figures 5.24 and 5.25, respectively. The flash time is 100 ms and the dark time 75 ms and for the ranking the median is used.

Three Subjects

For the group ranking the results of three different subjects were used. The first picture set (food) mean ranking is illustrated in Figure 5.26 and uses 100 ms for the flash time and 75 ms for the dark time. In comparison, Figure 5.27 shows the top 5 ranking with all the 4 different parameters but only for 10 flashes per image.

The baby picture set mean ranking has the same configuration as the food picture set and is illustrated in Figure 5.28. Furthermore, the ranking with all the different parameters (mean, mean + MAD, median and median + MAD) is depicted for only 10 flashes per images, see Figure 5.29.



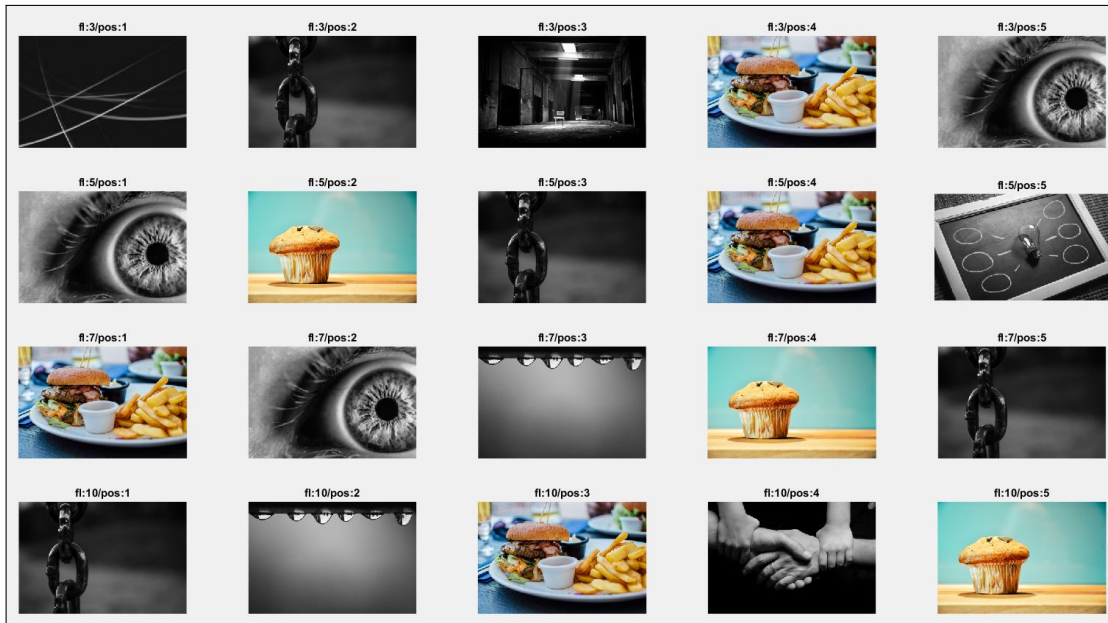


Figure 5.24: Top 5 images for different number of flashes with 100 ms for the flash time and 75 ms for the dark time and median as the ranking parameter.

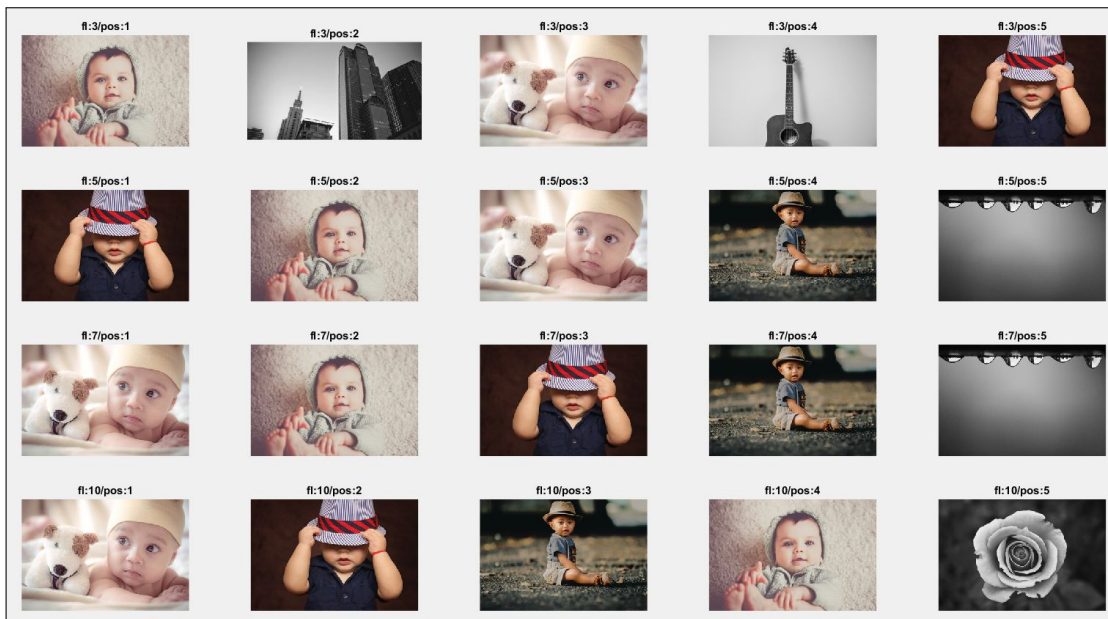


Figure 5.25: Top 5 images for different number of flashes with 100 ms for the flash time and 75 ms for the dark time and median as the ranking parameter.

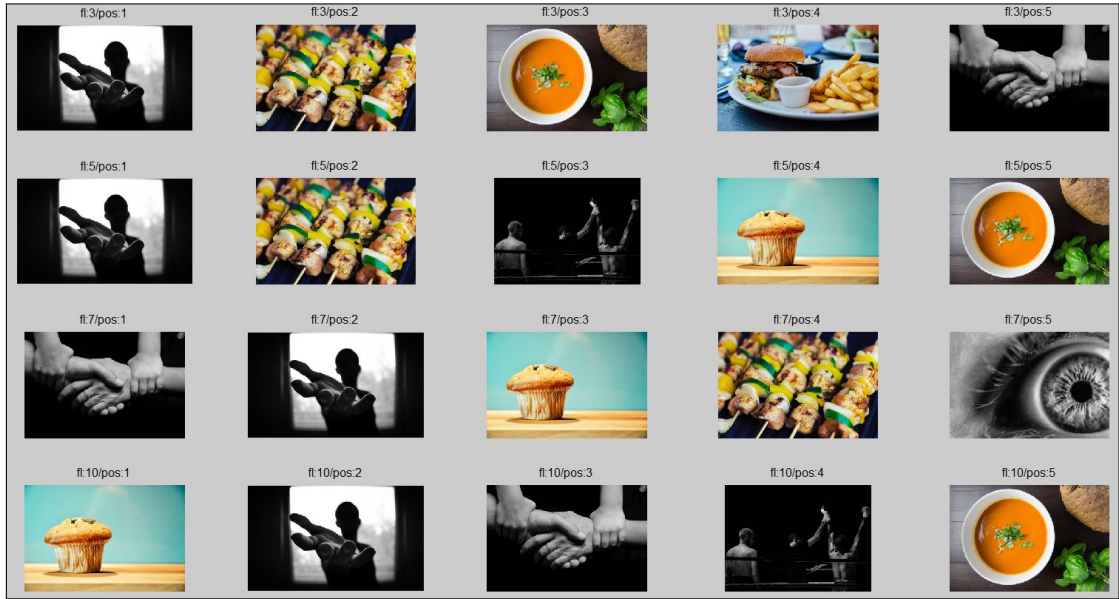


Figure 5.26: Top 5 images for different number of flashes with 100 ms for the flash time and 75 ms for the dark time and mean as the ranking parameter.

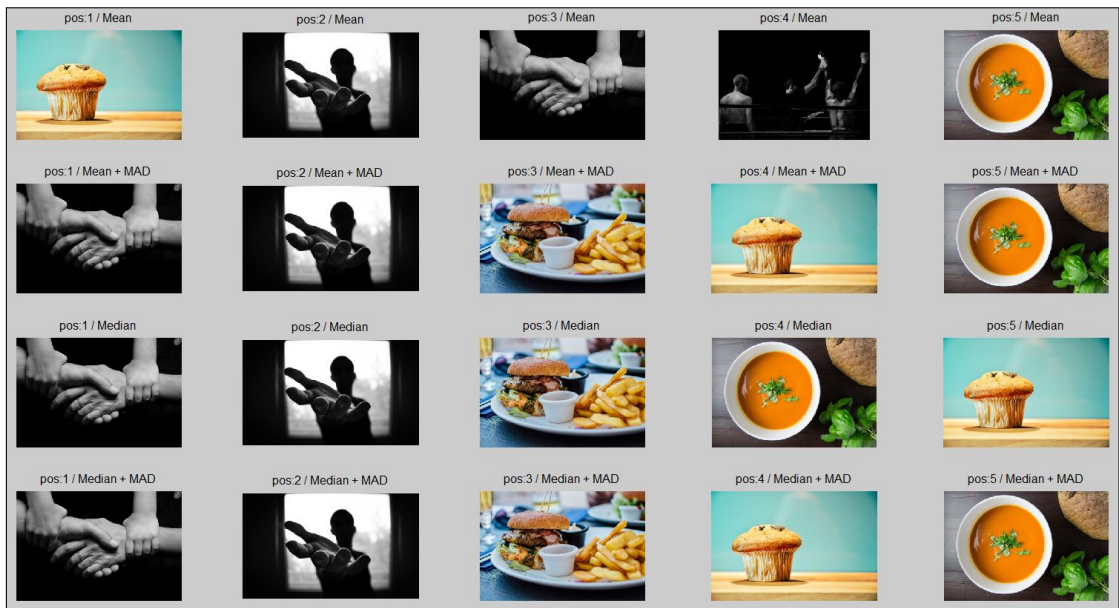


Figure 5.27: Top 5 images for 10 flashes with 100 ms for the flash time and 75 ms for the dark time and different ranking parameters.

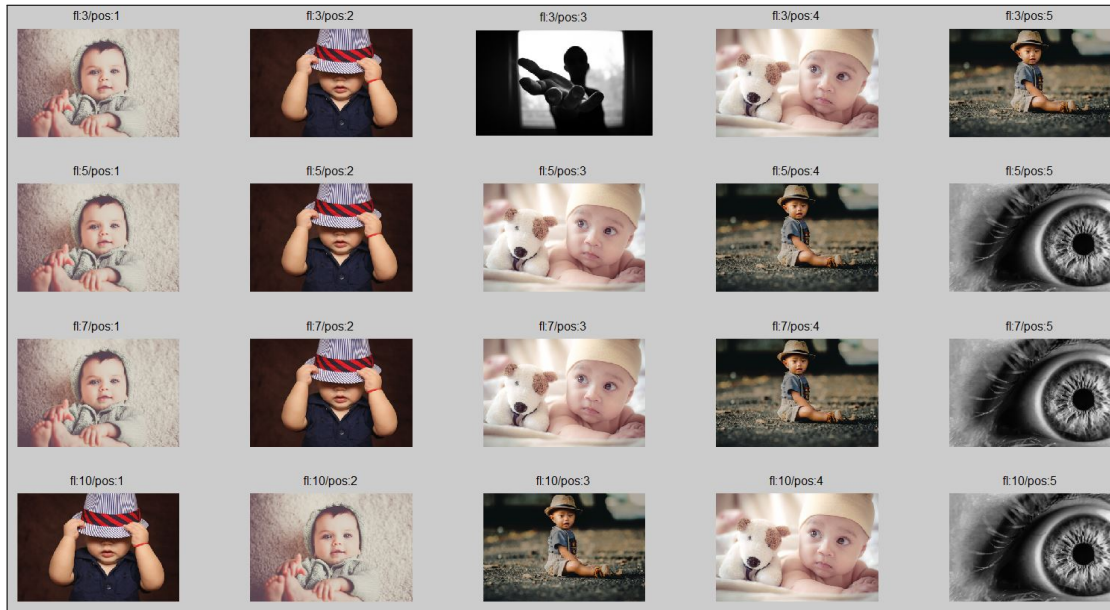


Figure 5.28: Top 5 images for different number of flashes with 100 ms for the flash time and 75 ms for the dark time and mean as the ranking parameter.

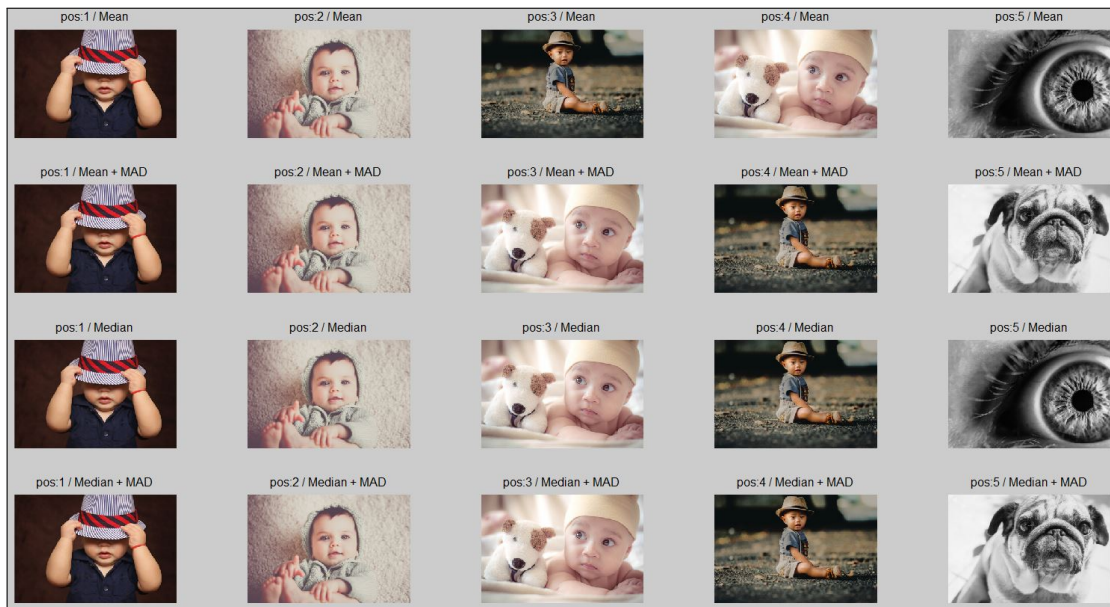


Figure 5.29: Top 5 images for 10 flashes with 100 ms for the flash time and 75 ms for the dark time and different ranking parameters.

Discussion

Here, the results shown in the previous section are explained and will highlight expected or unexpected behaviour.

6.1 Sampling Rate

The odd behaviour at the beginning of the recordings (Figures 5.2 and 5.4), where the sampling rate seems to be faster for the first few seconds, can be contributed to the initialisation time of the application. At the moment the application starts the device starts to record, but due to the initialisation the recorded data will be saved into a buffer. This buffer unloads the saved data, after the initialisation is done, which will be faster than samples are generated. Therefore, the sample rate calculations (table 5.1), where the first 1500 samples were cut off, negate this behaviour and only data without this interferences were considered. Also, the filters have a settling time of 20 - 30 seconds, which further indicates that the first few seconds are not relevant.

When examining the results of table 5.1 of the sampling frequencies are around the defined values of 250 Hz for the g.Unicorn and 256 Hz for the g.USBamp (considering the shortened data records). These values indicate that the time difference between the optimal values and the measured values for each sample differ and yield to a negative drift for both devices, see Figures 5.5 and 5.6. But this drift only has an effect for rather long recordings and even then they can be ignored, because for the calculations with the EEG signal only short windows of 800 ms length are cut out. For these short time intervals the drift of the devices can be ignored. Moreover, the values for the drift are illustrated in Figures 5.9 and 5.10 for the complete recording and the recording with the first 1500 samples cut off, respectively.

Figures 5.7 and 5.8 show the time between each sample for both devices. The left plot for each Figure shows all values, while the right one is a zoomed version. In this zoom version it becomes more apparent that there is often no time difference between samples, this can be attributed to the fact that the smallest time interval for the actual time recording is only 1 millisecond. For both devices the mean time between samples was calculated and equals the expected time, 4 ms for the g.Unicorn and 3,9 ms for the g.USBamp.

6.2 Trigger Display

Identification of the time between trigger onset in application and the illustration of this trigger image on the display screen is of importance to make sure the trigger response in the EEG is actually in the trigger frame. Figure 5.11 depicts this jitter of the trigger for both devices. Furthermore, the different timings for ISI, flash time and dark time are shown in Figures 5.12, 5.13 and 5.14, respectively. For the g.USBamp the time between triggers (Figure 5.12) is always the same, that is why only a vertical line a 312,5 ms is depicted. Which indicated that the number of samples is always the same too.

In tables 5.2 and 5.3 the calculated results for both devices are listed. The mean jitter of the trigger for all tested combinations of flash and dark time 57,9 ms for the g.Unicorn and 46,96 ms for the g.USBamp. This time probably originates from the different stages the command for loading an image as well as displaying it has to pass.

Moreover, this time difference will not effect the functionality of the application, because the trigger frame for a trigger response is 700 ms long and the most important component, the P300, occurs around 300 ms to 500 ms. The jitter moves the response to a later time where it is still within the trigger frame. Flash time and dark time over all the different combinations are for both devices about 12 ms shorter and about 30 ms longer, respectively.

These time changes should not impact the performance, the only problem which may arise is the flash time difference. In several recordings with the two fastest ISI times (50 ms/50 ms and 75 ms/75 ms) of the g.Unicorn device trigger images will not be displayed on the screen. The loss rate for the fastest configuration is about 3% and for the second fastest smaller than 1%. This loss rates are not of impact because they are negated with the averaging of all responses. Furthermore, the probability of losing the same trigger twice is minuscule. Other methods would be to select a higher number of flashes, use the g.USBamp or another configuration for ISI.

6.3 Trigger Frame

The LDA classifier parameter (weights and bias) calculation need all the signal processing steps as well as the feature matrix generated and the LDA calculation. Therefore, comparing weights and bias of a copy spelling run allows for the inspection of the provided g.tec scripts and the self scripted functionality.

If there are no differences, than that means that the number of samples for pre- and post-trigger are correct as well as the signal processing steps (downsample, baseline correction, moving average filter, downsample again). The feature matrix generation process is as expected. Lastly, the steps for the LDA classification, which are the most complex ones are also done right.

6.4 Picture Set

Figures 5.16 and 5.17 show the calculated accuracy for the different number of flashes. These and the ones done in table 5.4 with the script developed by g.tec. Generally speaking, all tested picture sets in colour or in grayscale generate a accuracy of 100% with only one flash.

This result seems strange, but several picture sets were tested. For these an LDA classifier was created for each and then tested in the copy spelling mode. Interestingly, the target was detected. But not all picture sets were tested in this fashion.

6.5 Copy Spelling

For the copy spelling mode, an LDA classifier had to be created before the test could start. The used picture set contained 35 grayscale images and one coloured image of a baby. Moreover, for each ISI combination and the classifier creation 30 flashes per image were selected. In [15], [31] and [12] images of faces generated a more pronounced ERP response, for that reason a image from a baby was chosen.

The different configurations for the flash time and dark time were derived from [26] where the ISI was 175 ms and in combination with [12]. Afterwards, the 100 ms/75 ms configuration as well as the 75 ms/100 ms one were chosen, the selected time went down to 50 ms/50 ms in 25 ms steps. The step size seemed appropriate, because it allowed for a noticeable change in the configuration. 50 ms/50 ms is the lower bound after a short test run with 25 ms for flash or dark time was way to fast. The upper bound and the gap between the highest ISI and second highest ISI were selected on the basis that 150 ms/150 ms was the initial configuration. Moreover, the gap resulted from the already good accuracy values for the 100 ms/75ms combination, started collection of data from the shortest ISI and made it longer.

Subsequently, for the copy spelling 5 target images of a new picture set, with 36 coloured images, had to be picked. Each run configuration (ISI combinations and number of flashes) was executed five times. These five images were defined at the beginning at random and were used for all runs. All runs were conducted by one subject over the span of several weeks.

Furthermore, a comparison between the results in table 5.5 and the ones from [26], [18] and [21] conform that the accuracy gets higher with a longer ISI as well as that at a certain point a maximum is reached. It also meets the assumption a higher number of flashes increases the accuracy. Additionally, a rather good accuracy could also be detected with a lower amount of flashes for each image. The reason behind that may be the concentration of the subject dropped with more flashes and could only be negated by more number of flashes.

In addition, an accuracy for the five target images was created as well, see Figure 5.19. It shows a much better accuracy rate for the image with the baby, which also complies with the observations in [15], [31] and [12].

6.6 Ranking

An LDA classifier had to be generated, for this a picture set of 35 grayscale images and 1 coloured images was used, where the coloured one was the target. This coloured image is of a baby, which should with the observations made in [15], [31] and [12] lead to a better classifier accuracy. The same configurations for the ISI timing were chosen as depicted in table 4.1 with 30 flashes for each picture.

For the ranking two new picture sets were created were both had the same 31 grayscale images, but differed in the 5 coloured images. A food and a baby picture set were created were

the coloured images 5 different kinds of food depict and 5 different baby pictures. For each picture set 5 runs were executed for each ISI configuration with 20 flashes per image.

Furthermore, the top 5 images were analysed for all recordings for one flash time and dark time configuration. Additionally, the results also show the different number of flashes, 3, 5, 7, 10, 12, 15 and 20. All the observations show that not only the coloured images are in the top 5 images. The pictures of a baby do occur more often in the top 5 then the one for the different kinds of foods. This result supports the examined behaviours in the studies [15], [31] and [12].

6.7 Group Ranking

After analysing the results from section Copy Spelling and Ranking the selected configuration used was the one with a flash time of 100 ms, a dark time of 75 ms and either 20 flashes for LDA classifier creation or 10 for the ranking. Additionally, the same picture sets as the ones mentioned in section Ranking were used for classifier creation and ranking.

This setup is faster to the initial configuration with a factor of about 1,2 and a higher accuracy for the application. This enhances the usability of the applications and lowers the resistance for usage.

For the group ranking of all the different subjects four different ranking parameters were created: mean, mean + MAD, median and median + MAD. Figures 5.27 and 5.29 show the top 5 images for each parameter with 10 flashes per images. Moreover, all but one ranking parameter yield the same result for the top 3 and have only position changes for the 4th and 5th place.

The only parameter with quite different results are quite different is the mean of the pictures position. This is an expected behaviour, because for the mean parameter, outliers have the same weight as any other values. In contrast, the median is more robust against outliers. The added MAD parameters are depending on the main parameter and calculate the mean or median absolute derivation to the mean or median parameter. respectively. These two MADs are needed if the mean or median ranking score is the same for two different pictures.

Conclusions

The evaluations for the different parameters, showed that the two devices as well as the applications functionality are working as intended. As already mentioned the smallest interval of the time logging data was 1 millisecond, this could be changed to make the increment smaller.

Furthermore, the best configuration of flash time, dark time and number of flashes would be 100 ms for the flash time, 75 ms for the dark time and 20 flashes for the classifier creation and 5 flashes for the ranking. These copy spelling results are a good basis to lean on for a more detailed analysis of the parameter combinations. In fact, the results are only generated by one subject and should be therefore, extended with more subjects. However, this configuration is 1,2 times faster than the initial one and has a higher accuracy.

Moreover, additional flash time and dark time combinations could be added and compared if they achieve a better accuracy. Additionally, smaller or bigger picture sets could be tested, but every change in picture set size may change the accuracy. The smaller set changes the frequency of the appearance of the trigger and thus the elicited P300 is not as significant. In contrast, a bigger picture set could lead to more fatigue during the application runs.

The execution of *Copy Spelling*, *Ranking*, and *Group Ranking* (for one person) together takes about 90 hours. In this time period the possibility of retaken test run again are not included. The one group ranking run takes about 1,5 hours for both picture sets (baby and food).

For the ranking of all the subjects the recommended parameter is the median in combination with the median absolute deviation. The group ranking parameter could be improved to be more robust against outliers and identify rankings were the user did not use a classifier or a bad classifier. For now the operator/user has to ensure that only data with a good classifier were considered.

Additionally, more detailed statistics could be generated were the picture could be grouped ontologically. This could provide a better understanding which group would achieves a higher ERP response.

The electrodes cap used needed gel to measure good EEG signal. A possible way to enhance the usability even more could be the development of electrodes which have no need for the gel and are not easily influenced by the environment.

Lastly, I want to mention that during the phase were I did most of the test run I had an unusual amount of time a headache. It may mean nothing but to stare repeatably for about 100 hours one a screen with flashing images could have been the reason. Especially, since the headaches stop after my test runs.

List of Figures

2.1	Structure of a neuron [1].	6
2.2	An example of an action potential [30].	7
2.3	Membrane potential with the influences of IPSPs and EPSPs [17].	8
2.4	ERP with its components [4].	8
2.5	ERP waveform in a visual oddball paradigm [19].	9
2.6	Electrode positioning of the electrode cap for both devices, g.Unicorn and g.USBamp [25].	11
2.7	BCI devices used for this thesis.	12
4.1	The two different setups to measure trigger timing on a notebook screen. <i>Setup 1</i> is with the g.USBamp and <i>Setup 2</i> uses the g.Unicorn	23
4.2	The target images for the new picture sets, each image is the target in one new picture set.	25
4.3	Target image for classifier creation.	27
4.4	The remaining images of the picture set for the <i>Copy Spelling</i> mode for LDA classification creation	30
4.5	The picture set used in the <i>Copy Spelling</i> mode for ISI analysis.	31
4.6	The five target pictures used for each run in the <i>Copy Spelling</i> mode.	32
4.7	The five coloured images of the <i>food</i> picture set for the ranking mode	32
4.8	The remaining images of the picture set for the <i>Ranking</i> mode	33
4.9	The five coloured images of the <i>baby</i> picture set for the ranking mode	34
5.1	Time elapsed for all samples for the g.Unicorn and a duration length of 5 min. . . .	36
5.2	Time elapsed for the first 2500 samples for the g.Unicorn and a duration length of 5 min.	36
5.3	Time elapsed for all samples for the g.USBamp and a duration length of 5 min. . .	37
5.4	Time elapsed for the first 2500 samples for the g.USBamp and a duration length of 5 min.	37
5.5	Time difference for each sample between optimal and measured time values for the g.Unicorn and a length of 5 min.	38
5.6	Time difference for each sample between optimal and measured time values for the g.USBamp and a length of 5 min.	39
5.7	Time difference between each sample of the measured data of the g.Unicorn with a duration length of 5 minutes.	40
		61

5.8	Time difference between each sample of the measured data of the g.USBamp with a duration length of 5 minutes.	41
5.9	Time difference between optimal and measured time values for each sample.	41
5.10	Time difference between optimal and measured time values for each sample beginning with the 1501st value.	42
5.11	Time delay between trigger onset in application and presentation on computer display.	42
5.12	Elapsed time between two triggers measured on the computer display.	43
5.13	Length of flash time on computer screen.	43
5.14	Dark time measured on the computer screen.	44
5.15	Results of the LDA classifier parameters (weights and bias) with the two different approaches for one copy spelling run.	45
5.16	Accuracy for the coloured baby target image, first run.	45
5.17	Accuracy for the coloured baby target image, first run.	46
5.18	Accuracy for the different configurations of the copy spelling settings.	47
5.19	Identification of the different target images in the copy spelling mode.	48
5.20	Top 5 images of the food picture set with different numbers of flashes and a flash time and dark time of 150 ms with the mean as the ranking parameter.	48
5.21	Top 5 images of the food picture set with different numbers of flashes and a flash time and dark time of 150 ms with the median as the ranking parameter.	49
5.22	Top 5 images of the baby picture set with different numbers of flashes and a flash time and dark time of 150 ms with the mean as the ranking parameter.	50
5.23	Top 5 images of the baby picture set with different numbers of flashes and a flash time and dark time of 150 ms with the median as the ranking parameter.	50
5.24	Top 5 images for different numbers of flashes with 100 ms for the flash time and 75 ms for the dark time and median as the ranking parameter.	51
5.25	Top 5 images for different number of flashes with 100 ms for the flash time and 75 ms for the dark time and median as the ranking parameter.	51
5.26	Top 5 images for different numbers of flashes with 100 ms for the flash time and 75 ms for the dark time and mean as the ranking parameter.	52
5.27	Top 5 images for 10 flashes with 100 ms for the flash time and 75 ms for the dark time and different ranking parameters.	52
5.28	Top 5 images for different number of flashes with 100 ms for the flash time and 75 ms for the dark time and mean as the ranking parameter.	53
5.29	Top 5 images for 10 flashes with 100 ms for the flash time and 75 ms for the dark time and different ranking parameters.	53

List of Tables

4.1	The different ISI configurations used.	26
5.1	The calculated sample rates for both configuration 1 and 2, with different duration lengths, which use the two devices g.Unicorn and g.USBamp,respectively	39
5.2	The measured time intervals for the g.Unicorn with different configurations. The values of last three rows in the table are relative to their configuration times.	40
5.3	The measured time intervals for the g.USBamp with different configurations. The values of last three rows in the table are relative to their configuration times.. . . .	44
5.4	Number of flashes needed to generate an accuracy of 100 % for the picture sets with different target images.	46
5.5	Accuracy table with all values in percentages for the different configurations.	47

Bibliography

- [1] APPsychology.com. <http://www.apppsychology.com/Book/Biological/neuroscience.htm>, March 2018. Accessed: 2018-03-27.
- [2] John M. Bekkers. Pyramidal neurons. *Current Biology*, 21(24):R975, December 2011.
- [3] K. Selçuk Candan and Maria Luisa Sapino. *Classification*, page 297–326. Cambridge University Press, 2010.
- [4] Yu-Kai Chang. Chapter 5 - Acute Exercise and Event-Related Potential: Current Status and Future Prospects. In Terry McMorris, editor, *Exercise-Cognition Interaction*, pages 105 – 130. Academic Press, San Diego, 2016.
- [5] Giorgio Corani, Joaquín Abellán, Andrés Masegosa, Serafin Moral, and Marco Zaffalon. *Classification*, chapter 10, pages 230–257. Wiley-Blackwell, 2014.
- [6] Michelle de Haan. Event-Related Potential (ERP) Measures in Visual Development Research. In Louis A. Schmidt and Sidney J Segalowitz, editors, *Developmental Psychophysiology*, pages 103–126. Cambridge University Press, Cambridge, 2007.
- [7] Richard O Duda, Peter E Hart, and David G Stork. *Pattern classification*. John Wiley & Sons, 2012.
- [8] Gunter Edlinger, Clemens Holzner, Christoph Guger, C. Groenegress, and Mel Slater. Brain-computer interfaces for goal orientated control of a virtual smart home environment. pages 463–465. IEEE Publishing, April 2009.
- [9] David Friedman, Yael M. Cycowicz, and Helen Gaeta. The novelty p3: an event-related brain potential (erp) sign of the brain’s evaluation of novelty. *Neuroscience & Biobehavioral Reviews*, 25(4):355 – 373, 2001.
- [10] g.tec medical engineering GmbH. <http://www.gtec.at/Products/Hardware-and-Accessories/g.USBamp-Specs-Features>, April 2018. Accessed: 2018-04-06.
- [11] Christoph Guger, Brendan Allison, and Junichi Ushiba. *Brain-Computer Interface Research: A State-of-the-Art Summary* 5, pages 1–6. Springer International Publishing, Cham, 2017.

- [12] Christoph Guger, Rupert Ortner, Slav Dimov, and Brendan Allison. A comparison of face speller approaches for p300 bcis. pages 004809–004812. IEEE, October 2016.
- [13] Christian Herff, Adriana de Pesters, Dominic Heger, Peter Brunner, Gerwin Schalk, and Tanja Schultz. *Towards Continuous Speech Recognition for BCI*, pages 21–29. Springer International Publishing, Cham, 2017.
- [14] Jing Jin, Brendan Z. Allison, Eric W. Sellers, Clemens Brunner, Petar Horki, Xingyu Wang, and Christa Neuper. Optimized stimulus presentation patterns for an event-related potential EEG-based brain–computer interface. 49(2):181–191.
- [15] T Kaufmann, S M Schulz, C Grünzinger, and A Kübler. Flashing characters with famous faces improves ERP-based brain–computer interface performance. 8(5):056016.
- [16] R. D. Keynes and D. J. Aidley. *Structural organization of the nervous system*, page 1–10. Studies in Biology. Cambridge University Press, 3 edition, 2001.
- [17] R. D. Keynes and D. J. Aidley. *Synaptic transmission in the nervous system*, page 103–117. Studies in Biology. Cambridge University Press, 3 edition, 2001.
- [18] Jessica Lu, William Speier, Xiao Hu, and Nader Pouratian. The effects of stimulus timing features on p300 speller performance. 124(2):306–314.
- [19] Steven J. Luck. Event-related potentials. In *APA handbook of research methods in psychology, Vol 1: Foundations, planning, measures, and psychometrics.*, pages 523–546. American Psychological Association, Washington, DC, US, 2012.
- [20] J N Mak, Y Arbel, J W Minett, L M McCane, B Yuksel, D Ryan, D Thompson, L Bianchi, and D Erdogmus. Optimizing the p300-based brain–computer interface: current status, limitations and future directions. 8(2):025003.
- [21] Dennis J. McFarland, William A. Sarnacki, George Townsend, Theresa Vaughan, and Jonathan R. Wolpaw. The P300-based brain–computer interface (BCI): Effects of stimulus rate. *Clinical Neurophysiology*, 122(4):731–737, April 2011.
- [22] Helmut Pfützner. *Angewandte Biophysik*. Springer, Wien, 2003.
- [23] Brian D. Ripley. *Linear Discriminant Analysis*, page 91–120. Cambridge University Press, 1996.
- [24] Saeid Sanei, J.A. Chambers, Saeid Sanei, and J.A. Chambers. Event-Related Potentials. In *EEG Signal Processing*, pages 127–159. John Wiley & Sons Ltd., 2007.
- [25] Saeid Sanei, J.A. Chambers, Saeid Sanei, and J.A. Chambers. Introduction to EEG. In *EEG Signal Processing*, pages 1–34. John Wiley & Sons Ltd., 2007.
- [26] Eric W. Sellers, Dean J. Krusienski, Dennis J. McFarland, Theresa M. Vaughan, and Jonathan R. Wolpaw. A p300 event-related potential brain–computer interface (BCI): The effects of matrix size and inter stimulus interval on performance. 73(3):242–252.

- [27] Siuly Siuly, Yan Li, and Yanchun Zhang. *EEG Signal Analysis and Classification*. Health Information Science. Springer International Publishing, Cham, 2016.
- [28] Samuel Sutton, Margery Braren, Joseph Zubin, and E. R. John. Evoked-potential correlates of stimulus uncertainty. *Science*, 150(3700):1187–1188, 1965.
- [29] Alaa Tharwat, Tarek Gaber, Abdelhameed Ibrahim, and Aboul Ella Hassanien. Linear discriminant analysis a detailed tutorial. *AI Communications*, 30(2):169–190, 2017.
- [30] Wikipedia.org. https://en.wikipedia.org/w/index.php?title=Action_potential, April 2018. Accessed: 2018-04-10.
- [31] Seul-Ki Yeom, Siamac Fazli, Klaus-Robert Müller, and Seong-Whan Lee. An efficient erp-based brain-computer interface using random set presentation and face familiarity. *PLOS ONE*, 9(11):1–13, 11 2014.