# Adapting Brain Signals With Reinforcement Learning Strategies for Brain Computer Interfaces

**Adaptieren von Hirnsignalen mit Reinforcement Learning Strategien für Gehirn Computer Schnittstellen**

Master-Thesis von David Sharma aus Offenbach am Main
Tag der Einreichung:

1. Gutachten: Dr. Elmar Rueckert
2. Gutachten: Prof. Dr. Jan Peters
3. Gutachten: Dr. Ing. Moritz Grosse-Wentrup

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Adapting Brain Signals With Reinforcement Learning Strategies for Brain Computer Interfaces
Adaptieren von Hirnsignalen mit Reinforcement Learning Strategien für Gehirn Computer Schnitt-stellen

Vorgelegte Master-Thesis von David Sharma aus Offenbach am Main

1. Gutachten: Dr. Elmar Rueckert
2. Gutachten: Prof. Dr. Jan Peters
3. Gutachten: Dr. Ing. Moritz Grosse-Wentrup

Tag der Einreichung:

# Erklärung zur Master-Thesis

Hiermit versichere ich, die vorliegende Master-Thesis ohne Hilfe Dritter nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den 26. Januar 2017

(David Sharma)

# Abstract

A problem of todays brain computer interface (BCI) systems is that performance in controlling a BCI can decrease rapidly over time. This is due to the non-stationarity of recorded electroencephalography (EEG) signals. Furthermore, the motivation of the subject can drop if the subject does not experience any success in controlling a BCI. A possible solution to these problems is to provide the subject with continuous feedback and to train a reinforcement learning (RL) agent on the task in order to support the subject in solving that task. A selection policy (implemented through a Monte-Carlo sampling process) selects either the command generated by the subject or by the RL agent. Especially in the beginning, the RL agent controls the actions of the task most of the time. As the experiment proceeds, the impact of the agent decreases and the subject gets more own control over the actions. The subject is not aware of the RL agent. To measure the performance of subjects, we implemented a scoring system, which rewards (positive or negative) the subject for its current performance, i.e., how good the subject solves the task.
We implemented a game, where the subject needs to control a game figure with the imagination of limb movements to jump over approaching obstacles. In our experiments, we collected data from 20 subjects. The evaluation of the gathered results, show a positive trend that subjects which trained with the reinforcement learning agent have a higher performance than subjects that did not train with the reinforcement learning agent. We also wanted to test if the subjects were able to adapt to new environments after the training. We first trained a classifier on the data from the training phase and used this classifier to decode new incoming EEG signals. We confronted the subjects to new obstacles. Unfortunately, performance of the subjects and the classifier were bad, such that we could not verify that the subjects were able to adapt to new environments.

# Zusammenfassung

Ein Problem heutiger Gehirn-Computer Schnittstellen ist, dass die Leistung eine Gehirn-Computer Schnittstelle zu kontrollieren, mit der Zeit schnell abnehmen kann. Das liegt daran, dass sich aufgenommene EEG Signale mit der Zeit verändern. Weiterhin kann die Motivation eines Probanden schnell nach unten gehen, wenn sich kein Erfolg einstellt beim Kontrollieren einer Gehirn-Computer Schnittstelle. Eine mögliche Lösung dieser Probleme ist, dem Probanden kontinuierlich Rückmeldung zu geben und einen Reinforcement Learning (RL) Agenten auf die Aufgabe zu trainieren, um dem Probanden beim Lösen dieser Aufgabe zu unterstützen. Eine Strategie, (implementiert durch eine Monte-Carlo Simulation) wählt dann aus, ob die Aktion vom Probanden oder die Aktion vom RL-Agenten ausgeführt wird. Vor allem am Anfang, werden die meiste Zeit die Aktionen des RL-Agenten ausgeführt. Mit voranschreiten des Experiments, nimmt der Einfluss des Agenten ab und der Proband bekommt immer mehr eigene Kontrolle über die Aktionen. Der Proband weiss nicht, dass ihn im Hintergrund ein RL-Agent unterstützt. Um die Leistung des Probanden zu messen, haben wir ein Punktesystem implementiert, welches den Probanden für seine momentane Leistung belohnt oder bestraft, also bewertet wie gut der Proband seine Aufgabe erfüllt.
Wir haben ein Spiel implementiert, wo der Proband eine Spielfigur allein mit der Vorstellung von Bewegungen seiner Extremitäten kontrollieren soll, um über sich nähernde Hindernisse zu springen. In unserem Experiment haben wir Daten von 20 Probanden gesammelt. Die Evaluation der gesammelten Daten zeigen einen positiven Trend, dass Probanden, die mit Hilfe des RL-Agenten trainieren, eine bessere Leistung zeigen, als Probanden, die nicht mit dem RL-Agenten trainieren. Wir wollten auch testen, ob die Probanden in der Lage waren, sich nach dem Training an neue Umgebungen anzupassen. Zuerst haben wir einen Klassifizierer auf den Daten vom Training trainiert. Danach benutzten wir den tranierten Klassifizierer, um neue EEG Signale zu klassifizieren. Wir konfrontierten die Probanden mit neuen Hindernissen. Leider war die Leistung der Probanden und des Klassifizierers schlecht, so dass wir nicht verifizieren konnten, dass die Probanden in der Lage waren, sich an neue Umgebungen anzupassen.

# Acknowledgments

I would like to thank my supervisors Elmar Rueckert, Daniel Tanneberg and Moritz Grosse-Wentrup for providing me with feedback and the discussions we had during this thesis. I especially want to thank Karl-Heinz Fiebig for all the discussions we had about my thesis. I also want to thank Tamara Friess, Natalie Faber, Thomas Hesse, Matthias Schultheis and Karl-Heinz Fiebig again for giving me feedback for my thesis and the good atmosphere in the D-Lab. I also want to thank all subjects, which took the time to participate in my experiments and all the helpers which helped to do the setup. Last but not least, thanks to Alex Blank, Marcel Musel and Friedrich Weber for providing me with more subjects for the experiments.

# Contents

# Figures and Tables

## List of Figures



## List of Tables

# 1 Introduction

People suffering from neurodegenerative diseases like amyothrophic lateral sclerosis (ALS) or tetraplegic persons have lost the ability to voluntarily move one of their limbs or even their whole body due to spinal cord injuries or degeneration of motor neurons. Motor neurons are responsible for controlling the muscles in the body [1].
Research in Brain computer interfaces tries to enable persons to control devices like exoskeletons, robotic arms or computer programs without the need to physically control their arms or feet, see Figure (1.1).

**(a)** Control of a robotic arm **(b)** Control of an exoskeleton **(c)** Control of a game

**Figure 1.1.:** There are different devices and computer programs that can be controlled with a brain computer interface. In a), a women controls a robotic arm with an invasive BCI, in b) a man tries to control an exoskeleton with a non-invasive BCI and in c) a man tries to play pong with a non-invasive BCI. Pictures taken from [2–4]

To control such devices or computer programs, brain signals need to be recorded. Recording brain signals could be done invasively by implanting microelectrodes on the surface of the brain or even in single cells to extract signals directly from the cells. The drawback of invasive techniques is that surgery is needed to implant the electrodes and that the electrodes need to be replaced after some time, because the immune response of the body damages the implanted electrode arrays. Non-invasive techniques do not need surgery to record activity in the brain. One of the most popular non-invasive techniques is the electroencephalography (EEG). Electrodes are placed on the scalp that are able to measure the electrical activity produced by millions of neurons. Further non-invasive techniques are magnetoencephalography (MEG) which the detects and measures magnetic fields of the brain, see Figure (1.2e)), functional magnetic resonance imaging (fMRI) which uses the blood oxygen level dependent (BOLD) signal [5] to display active areas in the brain, see Figure (1.2c)), functional near infrared (fNIR) imaging which uses near infrared light to detect the absorbance of light in blood with and without oxygen, see Figure (1.2d)), and positron emission tomography (PET) which uses a radioactive substance to detect metabolic activity, see Figure (1.2a)).

Brain computer interfaces combine knowledge and techniques from neuroscience, signal processing and machine learning. Besides having the hardware to record brain signals, it is also important to know which parts of the brain are responsible for certain mental processes and how the signals in the brain behave under these mental processes. Using mental processes to activate brain regions to control a device or a computer program is called paradigms. The most used paradigm is the motor imagery (MI) paradigm. Here, the subject imagines a movement with one of its limbs to activate areas in the sensory motor cortex (SMC). The activation in this area produces changes in certain frequency ranges which can be captured with signal processing techniques like the discrete Fourier transform (DFT). However, people suffering from ALS are not suitable for the MI paradigm since their motor neurons are damaged. Here, other mental paradigms need to be used, e.g., audio, mental subtraction, mental navigation, word association and mental rotation.

After using signal processing techniques to focus on certain regions and frequency ranges, features need to be extracted from the signals to classify them and translate them into different commands for the devices or computer programs. For this purpose, classifiers like linear discriminant analysis (LDA), support vector machines (SVM) or logistic regression are commonly used. These classifiers are able to recognize patterns in the signals and discriminate between them. Discriminating the signals makes it possible to translate these signals into different commands.

## 1.1 Motivation

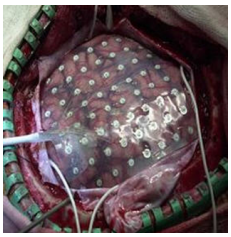
A problem with current BCI systems is that performance in controlling a BCI can decrease rapidly over time, since brain signals are non-stationary [6]. Performance in this work denotes the score the subject has achieved in solving the task

during a run. Another problem is that if no success in controlling a BCI is experienced, motivation of the subject can drop quickly [7]. To learn the necessary controls in a BCI system, the subject needs feedback to adapt to it and to judge its current performance. A potential solution to the aforementioned problem is that a training scheme is designed where the subject gets continuous feedback in form of score values displayed on a computer screen. With this feedback, the subject should adapt to slightly changing behavior of the device or computer program it is controlling.

Different feedback types have already been used in other experiments like biased feedback [8], real-time cortical activation maps [9], haptic robot-based feedback [10] and adaptive feedback [11]. To keep up the motivation of the subject, an optimal reinforcement learning (RL) [12] agent can be used which can control the device or program perfectly and knows how to behave in every situation. While getting much help in the beginning of the experiment, the help of the agent decreases with time, giving more control to the subject. The aim of combining an optimal agent with additional feedback is to evaluate, if subjects with RL assistance are able to learn better performing policies.

If the subjects in the RL group learn better performing policies than the subjects in the control group, it can be assumed that an increasing self-control for a task has an positive impact on the subject and thus supports it in learning to control a BCI system. Similar training schemes are used in BCI systems for monkeys [13], where the monkeys first control a device with real movements of the arm to extract optimal parameters for the movement. After their arm was restrained, an adaption algorithm adapts the parameters to the monkeys thoughts. The changing parameters reflect the change in the firing rates of the monkeys neurons. In this work, we evaluate this training concept in humans to see if humans are able to adapt to changing task behavior, from nearly optimal performance in the beginning with help of the RL agent to more own control over the task as the time proceeds. As far as we know, this was not done so far.

**(a)** PET

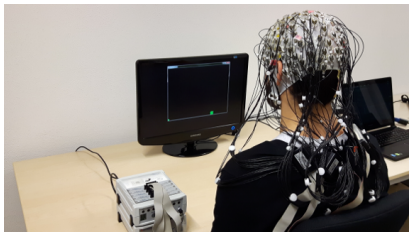


**(b)** ECoG

**(c)** fMRI

**(d)** fNIRS

**(e)** MEG



**(f)** EEG

**Figure 1.2.:** There are different possibilities to record brain signals. In a) a positron emission tomography (PET) device is shown, b) is the electrocorticography (ECoG), an invasive technique where electrodes are placed on the exposed surface of the brain, c) is a functional magnetic resonance imaging (fMRI) device, d) is a sketch of a functional near-infrared spectroscopy (fNIRS) setup, e) is a magneto-encephalography (MEG) device and f) shows a typical electroencephalography (EEG) setup. Pictures taken from [14–18]

## 1.2 Outlook

In Section 2, we briefly review related work. In Section 3, we give an overview about using the EEG hardware for recording signals, show how to process these signals and how to extract features of. The features are used to train classifiers. Common classifiers used by the BCI community are briefly introduced. Also the used reinforcement learning algorithms will be explained in this background section.
In Section 4, we will explain how we use reinforcement learning, Monte-Carlo sampling, Openvibe, an open-source software for BCI to build BCI pipelines, with the game to conduct the experiments.
In Section 5, we present the results of the training phase, where the subject needs to use motor imagery to control the game. Then, we present the results of the phase where subject needs to adapt to novel environments. For this purpose, a classifier is trained with the data from the training phase.
In Section 6, we conclude our results and discuss possible future work.

# 2 Related Work

**Overview**

Learning to control a BCI was not only done with humans but also with primates to analyze the impact of decreasing optimal control on primates [13]. Furthermore, the type of feedback which is provided to the subject plays an important role in terms of motivation and reducing training time. Here, we briefly want to summarize the work done with primates and used feedback techniques.

**Monkey BCI training**

In [13] primates learn to control the motion of a cursor in three dimensions with a BCI. Initially, the primates control the motion of the cursor, using arm movements (arm control mode). By using their arms to control the cursor, a mapping between arm movement and activity in the neuronal ensemble is created. This mapping is encoded in a population vector (PV). The PV fits the firing rates of each of the neurons to a linear function. The neuron with the highest firing rate for a preferred direction is encoded in a vector $(a_1, a_2, a_3)$ where $a_1, a_2$ and $a_3$ are the weighting coefficients of the linear function.
After the initial PV is estimated, the arm of the primate is restrained to solely control the cursor with its brain (brain control). Since the performance with the initial PV dropped after restraining the arm of the primate, the PV is adapted with an adaptive algorithm in a supervised fashion. The PV is adapted by changing the weighting coefficients and the contribution of each neuron to the population. The contribution of the neurons which were modulated during the attempt to control the cursor is increased, while the contribution of the neurons that were not modulated during the control of the cursor is decreased.
Finally a maximum likelihood algorithm (ML) is used to determine how reliably the firing of the neurons could be used to predict the movement to a target. In the arm control mode, the average prediction of the intended movement is around 65%. In early stages of the brain control mode, the average prediction drops to 35%. By adapting the parameters with the supervised learning control algorithm the average prediction increases to 80% in later stages of brain control mode.
The results show that a best possible estimate for the initial mapping between the activity of the neuronal ensemble and the cursor movement is not needed. Instead, the adaptive algorithm makes it possible to obtain a specific mapping between neuronal activity and cursor motion by changing the parameters in relation to changes in brain activity while learning to control the cursor without using physical movements.

In [19], the primates additionally learn to control a robotic arm with a BCI. The first step is to learn to control a cursor by moving it or to change the size of the cursor with a hand-held pole (pole-control mode). As the monkey tries to control the cursor, multiple linear models are trained to extract motor parameters like hand position, velocity, gripping force and multiple muscle electromyograms (EMGs) from the activity of neuronal ensembles in different brain areas.
The monkey is trained on three different task, a reaching task, a hand-gripping task and a reach-and-grasp task. Depending on the task, only some of the previously defined parameters was used. As the linear models converge to optimal performance, their coefficients are fixed and the activity of the neuronal ensembles are used as input for the linear models to predict the cursor movement or the change in cursor size (brain control mode).
After initial movements of the arm in brain control mode, the monkeys realize that the movements are not necessary and cease to produce them for periods of time. Later, the pole is removed completely. After initial training a 6 DOF robot arm equipped with a 1 DOF gripper is introduced for each task. The movement and the change of size from the cursor is mapped on to the robot arm and the gripper respectively. For each task, the monkeys are able to control the cursor and the robot arm with the gripper completely without the hand-held pole and only with the modulation of the signals from the neuronal ensembles. The size of the neuronal ensembles determines the performance of the primates. Larger neuronal ensembles contribute to better performances of the primates which shows that the used movements are highly distributed across brain areas.

**Human BCI training**

Another important part of this thesis is the type of feedback ,the subject is provided with. There already has been work that dealt with different kind of feedback types.
In [9] real-time brain activation maps are used to provide feedback. A time varying map of cortical rhythmic activity is shown to the subject, which updates itself every 350 ms. By imagining left or right hand movements, the subjects are able to activate the brain regions which are responsible for left or right hand movement. This activation is shown to the subject in real-time. After a while of training, the subjects are able to reach a stronger activation of the respective

brain areas and also the classification accuracies of the used classifiers increases. Providing the subject with real-time neurofeedback seems to be a good way to train the subject to learn to control a BCI since the subject immediately sees the activations in the respective brain regions and knows what kind of thoughts influence the activations.

Another promising type of feedback is the co-adaptive feedback, where the classifier tries to adapt to the subjects signals and the subject tries to adapt its thoughts to the outputs of the classifier.
In [11], co-adaptive feedback is used in the following way. In the first step, the subjects use a subject independent classifier which uses simple band power features to classify the subjects signals. The classifier is updated in a supervised fashion by using the signals from the past trial. In the second step, more complex features are used. The features of the first three runs are used to determine spatial filters with common spatial patterns (CSP) analysis. Furthermore, six Laplacian channels are selected according to their discriminability by using a robust variant of the Fischer score. Channel selection and the classifier are updated after each trial using the last 100 trials. The feature vector is the concatenation of the log bandpower in the CSP channels and the selected Laplacian channels. The update in this step is also done in a supervised fashion. The last step is to calculate CSP filters on the data of the second step and to train the classifier on the resulting log bandpower features. This step is done unsupervised, which means that no class labels are used. Subjects who previously were not able to develop an activation in the sensory motor-cortex (SMC), learned to activate the SMC with the co-adaptive approach.

Even though feedback seems to be important to learn to control a BCI, some subjects still are not able to learn to control a BCI and thus may loose motivation in learning the task which could influence the subject negatively.
In [8], the authors investigate, how biased feedback influences the subject in learning to control a BCI. The subjects belief in controlling the BCI is biased negatively or positively in 80% of the trials, which means that the feedback is randomly distorted. The authors speculate that subjects already capable of controlling a BCI are negatively influenced by biased feedback while subjects performing close to chance level may be positively affected by biased feedback. Thus, the subjects skill level should be taken into account when providing the subject with feedback.

The most used feedback type for BCIs is the visual feedback but other types of feedback are also possible, like haptic feedback for example.
In [10] a BCI is combined with a robotic arm, attached to the subjects forearm. The idea of this approach is to develop a method to help stroke patients to restore the disrupted sensorimotor feedback loop. For that purpose, a BCI is used to decode the subjects movement intentions and the attached robot arm is used provide haptic feedback. Based on the classifiers output, the robot arm moves the arm of the subject forward or backward. Synchronizing the movement intention and robot-assisted physical therapy, seems to result in increased cortical plasticity due to Hebbian-type learning rules. The results suggest, that closing the sensorimotor loop through haptic feedback improves the decoding of movement intentions in healthy subjects. Furthermore, the subjects are able to modulate their sensorimotor rhythm (SMR).

# 3 Background

Here, we present the basic knowledge about EEG signal processing, EEG feature extraction. Furthermore, we present common used classifiers for BCI and introduce reinforcement learning.

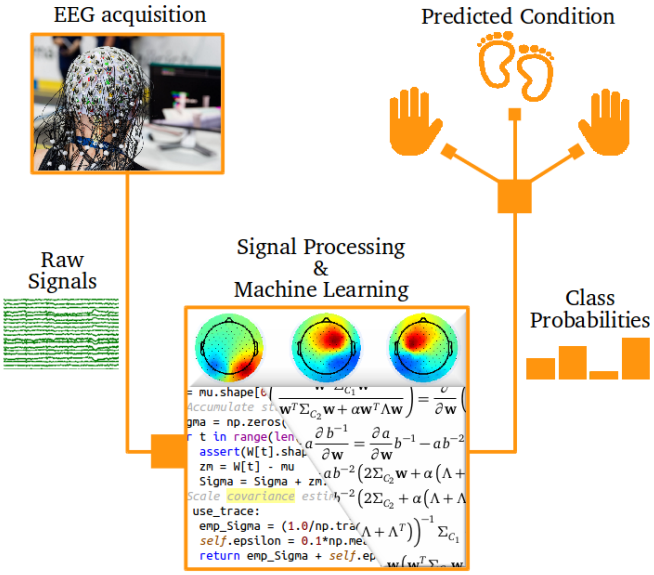



**Figure 3.1.:** Sketch of a simple BCI pipeline. After acquiring the raw signals, different signal processing techniques have to be applied to extract features. The features are used to train classifiers. Based on the trained classifier model, the classifier tries to predict different conditions, using the motor imagery (MI) paradigm. Picture taken from [20].

## 3.1 Electroencephalography for BCIs

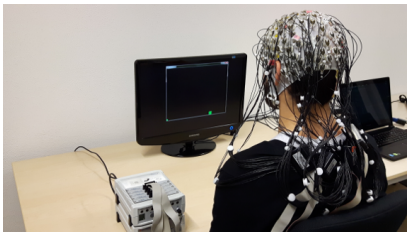
### EEG hardware

In this thesis, an Electroencephalograph (EEG) is used to record brain signals. The EEG is a non-invasive technique to record brain signals. Typically, the main components of an EEG are the electrodes, which are able to receive brain signals produced by populations of neurons and an amplifier, which needs to amplify the incoming signals because they have to go through three layers- the meninges, the skull and the skin. An EEG signal has a low spatial resolution because one electrode receives signals from millions of neurons but a good temporal resolution because the electrodes can capture changes in the signals on a millisecond scale. Furthermore, an EEG can only capture strong signals from the cerebral cortex. Signals deeper in the brain cannot be captured because voltage fields fall of with the square of the distance from the source [21] in gray matter as in air. Throughout this thesis an 128 electrodes actiCHamp system from BrainProducts[1] is used to do the experiments.

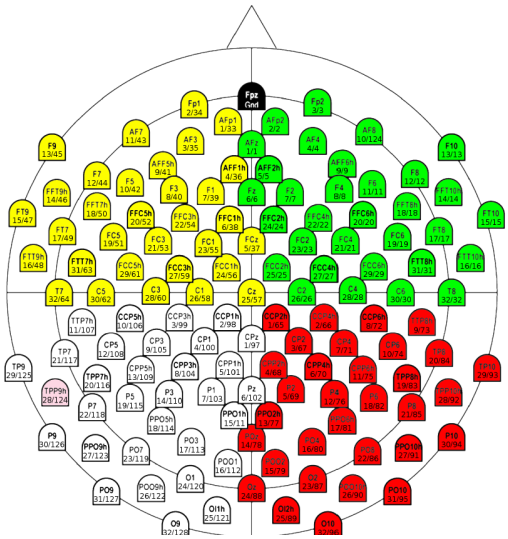
### EEG signal processing

The electrodes of an EEG are placed on the scalp of the subject to acquire signals from the brain which are produced by active populations of neurons. Since the spatial resolution of an EEG is low, the electrodes only can capture correlated activity by large populations of neurons, such as oscillatory activity [21].
By executing real movements or just imagining movements with hands or feet, the same areas in the sensory motor cortex (SMC) are activated. Using imagined movements to activate the SMC is called motor imagery (MI). To exploit this knowledge of activation in the SMC and analyze the EEG signals, they can be transformed from the time domain into the frequency domain, i.e., with discrete Fourier transform (DFT). Brain signals can be divided into different frequency bands, delta waves (0.5 - 3 Hz), theta waves (4 - 7 Hz), alpha/mu waves (8 - 13 Hz), beta waves (14 - 30 Hz) and

[1] http://www.brainproducts.com/



**(a)** EEG



**(b)** 128 electrode 10-20 system

**Figure 3.2.:** In a), person uses the actiCHamp sytem from Brain Products while trying to control a game with a BCI. In b), we show the 10-20 system 128 electrode placement.

gamma waves ($>$ 30 Hz). Delta waves are an indicator for deep sleep or deep unconsciousness, theta waves indicate transition between deep sleep and wakefulness, alpha waves indicate inactive wakefulness and relaxation, beta waves indicate active wakefulness, gamma waves indicate strong concentration and learning.
When executing or imagining a movement, the power of EEG signals changes in some frequency bands. The decrease of power in an EEG signal is called event related desynchronization (ERD). The increase of power in an EEG signal is called event related synchronization (ERS). By imagining a movement for example, the EEG signal desynchronizes in the alpha/mu and beta band when a movement is imagined and synchronizes in beta band after the imagination of the movement.

Raw EEG signals are noisy, non-stationary, complex and of high dimensionality [22]. Thus, to analyze the recorded EEG signals, they first need to go through different pre-processing steps. After pre-processing the signals, features will be extracted and used to train a classifier, to assign classes to different sets of features which encode the imagined movement from a person. Since EEG signals can be contaminated by physiological artifacts from muscles or environmental artifacts from electrical devices, different algorithms can be used to reduce those artifacts in order to acquire as clean brain signals as possible. Furthermore, if paradigms like MI are used, only certain electrodes over the SMC area and frequency bands in the alpha and beta range are of interest. To subtract common noise from different sources, enhancing local activity, reducing dimensionality and filtering frequency bands of interest, different temporal and spatial filters are used. Temporal filters like the Butterworth or the Chebychev filter are used to filter interesting frequency bands in the time domain and reject unwanted frequencies.
Spatial filters like the common average reference (CAR) or the Laplacian filter can be used to subtract common noise and enhance local activity. The CAR filter computes the average value over all electrodes and subtracts this average value from each electrode. Assume an EEG has $K$ electrodes and $x^{[i]}$ is the signal from electrode $i$. Then the CAR filter can be formalized as $\hat{x}_t^{[i]} = x_t^{[i]} - \bar{x}_t$, where $\bar{x}_t = 1/K \sum_{i=1}^{K} x_t^{[i]}$, and $\hat{x}_t$ is the corrected signal. A Laplacian filter computes the average from a set of electrodes $Q^{[i]}$, where $Q^{[i]}$ includes the four nearest neighboring electrodes from electrode $i$. The Laplacian filter can be formalized as $\hat{x}_t^{[i]} = x_t^{[i]} - \bar{x}_t^{(4)}$, where $\bar{x}_t^{(4)} = 1/4 \sum_{x_t^{[i]} \in Q^{[i]}} x_t^{[i]}$. For more details, we refer to [23].

#### EEG features for classification

After pre-processing the signals, the next step is to extract features for a classifier. There are different sources of information that can be used to extract features from EEG signals [24]. Spatial information is used to focus on certain areas of the brain, e.g., the SMC area for MI. Therefore, certain electrodes are selected that represent an area of interest. Spectral information represents frequency bands of interest, i.e., the change in power in the frequency bands. Temporal information represents the change of signal in time, i.e., looking at EEG values at different points in time, or even predefined time windows. For MI, mainly spatial and spectral information are of interest. For other EEG signals like event related potentials (ERP), temporal and spatial information are of interest. Since this thesis focuses on MI, further explanations

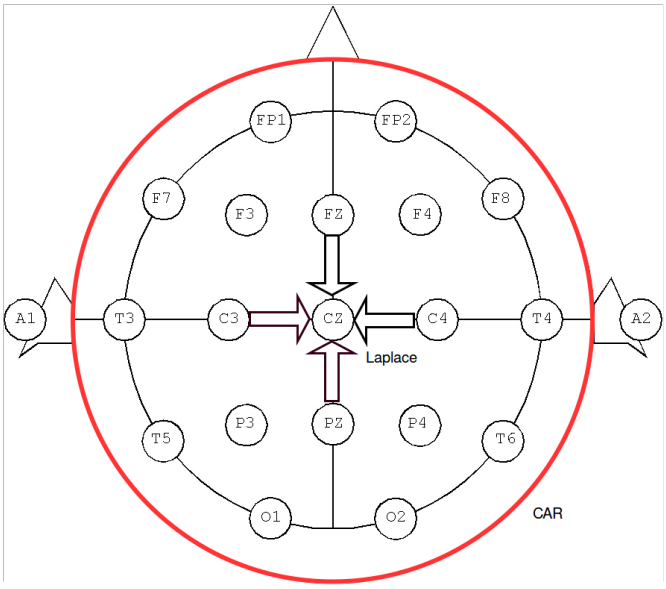



**Figure 3.3.:** Sketch to show the basic spatial filtering techniques. The red circle stands for the CAR filter which subtracts the average over all electrodes from each electrode. The Laplacian filter subtracts the average of the four nearest neighboring electrodes (arrows, pointing to the Cz electrode) from a certain electrode. In this sketch the average would be subtracted from Cz.

are based on spatial and spectral information.
To extract previously described information as features, after pre-processsing, the EEG data is transformed from the time domain into the frequency domain. To transform the signal in the frequency domain, discrete Fourier transform (DFT) is used. After the DFT is applied to the data, the log bandpower in the frequency bands of interest need to be calculated. Assume that $\mathbf{x} \in \mathbb{R}^n$ is an EEG time series with $n$ samples recorded from a single electrode. The DFT computes the spectrum over frequencies, denoted by $\omega_q = q2\pi/n$ for $q = 1, 2 \ldots, n$. The log bandpower of the frequency component $\omega_q$ can be computed from the absolute value of the complex spectrum, which is given by

$$\log V_{\omega_q}^2(\mathbf{x}) = \log \left| \sum_{k=1}^{n} x_k e^{i\omega_q k} \right| . \tag{3.1}$$

The log bandpower within a certain frequency range can be acquired by summing or averaging over the bandpower of all $\omega_q$ that fall into the frequency range.
Another possibility to calculate the log bandpower is the following. Here, the log bandpower of $\mathbf{x}$ can be calculated by applying a bandpass filter for a certain frequency range and use Parseval's theorem,

$$\log V^2(\mathbf{x}) = \log \sum_{k=1}^{n} |x_k|^2 = \log(\mathbf{x}^T \mathbf{x}),$$

where $x_k$ is the k-th element of $\mathbf{x}$. After the log bandpower features are calculated, classifiers can be trained on the extracted features to learn a model of the data. The features encode the change in power in a certain frequency due to an ERD/ERS. Common classifiers used in the BCI community are linear discriminant analysis (LDA), support vector machines (SVM) or logistic regression.

#### Linear discriminant analysis

LDA is a binary classifier. The goal of LDA is to find a hyperplane $h$ that separates two classes $c_1$ and $c_2$. The decision boundary then is characterized by the following equation

$$h(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0 ,$$

where $\mathbf{x}$ is the feature vector, $\mathbf{w}$ is the hyperplanes weight vector and $w_0$ is the bias of the hyperplane. By finding a good weight vector $\mathbf{w}$ and assuming the data is Gaussian distributed, LDA tries to maximize the class means $\mu_1$ and $\mu_2$, while minimizing the within class variances $\Sigma_1$ and $\Sigma_2$. By assuming that the class covariances are equal $\Sigma_1 = \Sigma_2 = \Sigma$, and have a full rank, we can obtain the solution for $\mathbf{w}$, i.e.,

$$\mathbf{w} = \Sigma^{-1}(\mu_1 - \mu_2) .$$

For a more detailed explanation on how to derive $\mathbf{w}$, we refer to [25]. A new prediction $y$ for a feature vector $x$ can be obtained by evaluating

$$y = \sigma(\mathbf{w}^T\mathbf{x} + w_0),$$

where $\sigma$ denotes the sign function.

#### Support vector machine

The hyperplane that LDA chooses is only one of a potentially infinite number of hyperplanes. While LDA already gives reasonably good results, it can be shown [26] that the best hyperplane is the one with the largest separation (margin) between two classes. A linear SVM finds such a hyperplane, where the margin between the two nearest data points (support vectors) of each class to the hyperplane is maximized. To allow misclassifications, slack variables $\xi$ for linear SVM's were introduced. However, misclassifications will also be penalized. A SVM with a slack variable is called soft margin SVM. The optimization problem for a soft margin SVM can be formalized as follows

$$\begin{aligned}
&\min_{\mathbf{w},\mathbf{b},\xi} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\xi_i\,,\\
&\text{s.t. } y_i(\mathbf{w}^T\mathbf{x_i} + b) \geq 1 - \xi_i\,,\\
&\xi_i \geq 0\,,\ i = 1,\ldots,l\,,
\end{aligned}$$

where $C$ is the penalty parameter.

#### Logisitc regression

While LDA and SVM use a decision boundary to predict classes, logistic regression uses probabilities to predict classes. It maps a data point $y$ in d-dimensional feature space to a probability value $p$. To do that, the sigmoid function, $p(z) = 1/1 + e^{-z}$ is used, where $z$ is the model to be learned. The model $z$ can be written as

$$z = \beta_0 + \beta_1 y_1 + \beta_2 y_2 + \cdots + \beta_d y_d\,,$$

where $\beta_0$ is the bias term and $\beta_1,\ldots,\beta_d$ are the coefficients. The $d$-dimensional vector $\beta$ has to be optimized in order to learn a good model for the data. Optimizing is usually done with the maximum likelihood estimation or gradient based optimizers.

## 3.2 Reinforcement Learning

Most of the parts in this section are taken from previous work I did in [27]. In reinforcement learning, an agent interacts in uncertain environments with the goal to maximize a numerical long term reward. In every state $s_t$, the agent can take an action $a_t$ to get to state $s_{t+1}$. The agent chooses its action according to a policy $\pi$, which maps actions to states. For every action, the agent gets feedback from the environment in form of numerical rewards $r_{t+1}$. To maximize the numerical long term reward, the agent has to explore its environment and update its policy to incorporate the knowledge into the policy about taking an action in a certain state. By maximizing the numerical long term reward, the agent learns how to act optimally in every state. Formally, a reinforcement learning setup consists of a possible set of states $\mathbf{S}$, a possible set of actions $\mathbf{A}$, a state transition function $\delta : S \times A \to S$, a reward function $r : S \times A \times S \to \mathbb{R}$ and a policy $\pi : S \to A$.

In this thesis, a discrete state space is used. It is also assumed, that the states have the Markov property. A state has the Markov property if it is able to summarize all past information in the state at time $t$, i.e., the transition function and the reward function are independent of past states. In mathematical terms, s state is a Markov state, if and only if,

$$P\{s_{t+1} = s', r_{t+1} = r | s_t, r_t, s_{t-1}, a_{t-1}, \ldots, r_1, s_0, a_0\} = P\{s_{t+1} = s', r_{t+1} = r | s_t, a_t |\}\,.$$

A reinforcement learning task that satisfies the Markov property is called a Markov decision process (MDP). This property makes it possible to predict the next state based on the current state and action without considering the history of all states and actions. In the following, we discuss how discrete MDPs can be solved.

### Value functions

A value function $V^\pi$ is an indicator for the agent of how much future reward it can expect for starting in a state $s$ and following the policy thereafter. The value function $V^\pi$ is also called the state-value function and can be formalized as

$$V^\pi(s) = E_\pi\{R_t|s_t = s\} = E_\pi\{\sum_{k=0}^{\infty}\gamma^k r_{t+k+1}|s_t = s\},$$

where $R_t$ is the long term reward specified by

$$R_t = \sum_{k=0}^{\infty}\gamma^k r_{t+k+1}.$$

The symbol $\gamma \in [0,1]$ denotes the discount factor which balances the importance of future rewards with right to the current one. If $\gamma$ is 0, the agent only considers current rewards. For $\gamma$ values close to one, future rewards become more important.
The state-value function can also be written in a recursive manner. Given a stochastic policy $\pi$ and a stochastic transition probability $P(s_{t+1} = s'|s_t = s, a_t = a)$, the equation can be written as

$$V^\pi(s) = \sum_a \pi(a|s)\sum_{s'} P(s'|s,a)(r(s,a,s') + \gamma V^\pi(s')).$$

This equation is called the Bellman equation for $V^\pi$. It expresses the relationship between the values of a state and the values of its successor states.
Another value function is called the state-action-value function $Q^\pi$. It is used to estimate, how much future reward the agent can expect by taking an action $a$ in state $s$ and following the policy thereafter. It can be formalized as

$$Q^\pi(s,a) = E_\pi\{R_t|s_t = s, a_t = a\} = E_\pi\{\sum_{k=0}^{\infty}\gamma^k r_{t+k+1}|s_t = s, a_t = a\}.$$

The recursive formulation of this is equation is

$$Q^\pi(s,a) = \sum_{s'} P(s'|s,a)(r(s,a,s') + \gamma V^\pi(s')).$$

With the Q-function, it is possible to evaluate the quality of transitions since it incorporates actions which are needed to do that. With the V-function, it is only possible to evaluate the quality of states which makes it more difficult to compute policies from it.

### Optimal value functions

A policy $\pi'$ is said to be better than policy $\pi$ if and only if $V^{\pi'}(s) \geq V^\pi(s)$, for all $s \in S$. If a policy is better than all the other possible policies, then this policy is said to be the optimal policy, denoted as $\pi*$. Formally, the optimal policy can be written as $\pi^* = \max_\pi V^\pi(s)$, for all $s \in S$. If the agent learned the best possible value for every state, it has learned an optimal value function denoted as $V^*$. The optimal value function can be written as $V^*(s) = max_\pi V^\pi(s)$, for all $s \in S$. Optimal policies also have optimal action-value functions, denoted as $Q^*$. The optimal action-value function can be written as $Q^*(s,a) = \max_\pi Q^\pi(s,a)$, for all $s \in S$ and $a \in A$. If the agent learned the optimal state-value function, then it always executes the best action in every state. The optimal value function is given by $V^*(s) = max_a Q^*(s,a)$ The Bellman optimality equation expresses that the expected return for the optimal value function under an optimal policy must be equal the expected value for the best action from that state. The Bellman optimality equation can be written as

$$V^*(s) = \max_{a \in A_s}\sum_{s'} P(s'|a,s)(r(s,a,s') + \gamma V^*(s')).$$

Similarly the Bellman optimality equation for the optimal Q-function can be written as

$$Q^*(s,a) = \sum_{s'} P(s'|a,s)(r(s,a,s') + \gamma \max_{a'} Q^*(s',a')).$$

Using these definitions, a widely used algorithm for solving discrete MDPs can be derived, which is shown next.

### Q-Learning

To execute optimal actions in every state, the agent has to learn optimal value functions for every state. One way to learn optimal value functions is called temporal difference (TD) learning. TD methods use old estimates of $V$ to generate new estimates of $V$ without need to wait for the final outcome. The agent updates its value functions immediately after it receives its rewards for a taken action based on the expected long term reward. The update function for $V$ can be written as

$$V(s_t) = V(s_t) + \alpha(r_{t+1} + \gamma V(s_{t+1}) - V(s_t)).$$

The parameter $\alpha$ is called the learning rate. If it is set to 0, the agent does not obtain any new information. If it is set to 1, the agent only considers most recent information. The term in the brackets is called the temporal difference error. It is the difference between the estimate of the value before and after performing the action.
In TD learning there are two different policy control methods, off- and on-policy control. One of them is Q-learning, an off-policy control method. Off-policy control methods use two different policies, one policy for behavior and one policy for estimation, i.e., while the agent chooses actions according to policy $\pi'$, it evaluates and improves the values for policy $\pi$. Q-learning estimates the value of the optimal policy (thus, no exploration). The update function for Q-learning is defined as

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha(r_{t+1} + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)).$$

Another method to update the value functions is the state-action-reward-state-action (SARSA) method, an on-policy control method. On-policy control methods evaluate and improve the same policy as they use to make their decisions. In SARSA, we estimate the value of the current policy (with exploration), however, as the value function defines the policy, the policy will change and also converge to the optimal policy. The update function for SARSA can be written a follows

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha(r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)).$$

Since Q-learning estimates the value of the optimal policy, it always chooses the best action. By always choosing the best action, we are likely to miss better actions in the long run because of lack of exploring. To avoid this greedy behavior, an additional policy to choose actions will be introduced next.

### Epsilon-Greedy policy ($\epsilon$-greedy)

Instead of acting greedy all the time, we act greedy most of the time, i.e., with a small probability of $\epsilon$, we choose an action at random. This policy is called the $\epsilon$-greedy policy. In this policy, the best action is selected with a probability of $1-\epsilon$. It can be formalized as follows

$$\pi(a|s) = \begin{cases} \frac{\epsilon}{|A|} & a \neq a^* \\ \frac{\epsilon}{|A|} + (1-\epsilon) & a = a^* \end{cases}.$$

In our experiments, we will use the Q-learning algorithm with an additional epsilon-greedy policy to train the RL agent which will be explained in the following section.

# 4 Guided training strategies for BCI (methods)

Here, we combine previously introduced methods to realize our approach of guided training strategies for BCI.

## 4.1 Overview

We give an overview about how our approach works and explain how single components work together. Furthermore, we use three Openvibe[1] pipelines to realize different phases of the experiment such as training of the subject, training of the classifier and testing the subject on previously unseen environments.

**The game**
To conduct our experiments, we implemented a simple game. In this game, the subject needs to control a figure where the figure needs to jump over approaching obstacles. The game has three phases. A relax phase, where the subject needs to relax, an active phase where the subject needs to imagine a movement with one of its limbs and a pause phase where the subject is allowed to move. Each of these phases take six seconds. In every frame of the game and every jump command in the active phase, the subjects gets -1 point, for staying on ground, the subject gets 0 points, for colliding with the obstacle, the subject gets -50 points and for succeeding to jump over the obstacle, the subject gets 400 points.

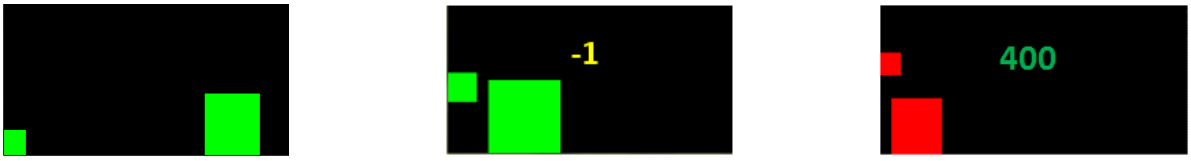



**Figure 4.1.:** The game was implemented in Python with the Pygame library. Here, the pictures show different states of the game and how feedback is provided by the scoring system.

**Signal pre-processing**
Before the EEG signals can be used to calculate features or sending actions to the game, they need to be pre-processed. To reduce common activity of neurons over all electrodes and focus on local activity, a common average reference (CAR) filter is applied, see Subsection (3.1). Then a 3-40 Hz bandpass Butterworth filter of fourth order is applied to the signal to reject unwanted frequency ranges. Only the signals in the six second time range of the active and relax phase are of interest. In order to extract features, we need to accumulate data in a time window first. In this case, we chose a data chunk (epoch) of three seconds and use it for the features. To make the system more reactive to the users intent, a 20 ms sliding window is being used over the epochs. This means that the game runs with 50 Hz. Since the values in the alpha frequency range are of interest to see if MI of the subject is successful, the log bandpower needs to be computed for this frequency range. First, the signal needs to be transformed from the time domain into the frequency domain. This needs to be done with the fast Fourier transform (FFT) [28], which is a more efficient way to transform the signal from the time domain into the frequency domain, compared to the discrete fourier transform (DFT). Before we convert the signal into the frequency range, a Hanning window is applied to the signal, to reduce discontinuities in the signal. To compute log band power features, we need to apply Equation (3.1) for every frequency component. Since we are interested in the log band power of the 8-13 Hz frequency range, we further need to compute the average (spectral average) over the band power of all frequency components that fall into the frequency range of 8-13 Hz. The resulting log band power values will be used as features for the classifier and for the signal-baseline difference to decide if a jump command will be sent to the game from the user.

**Signal-baseline difference**
To make the figure jump, the subject needs to use MI in the active phase. For MI, the subject has to imagine movements with one of its hands or feet. By imagining a movement, neurons in the respective sensory motor cortex (SMC) area desynchronize. This change in power is called event related desynchronization (ERD) or event related synchronization (ERS). The log band power for each electrode is computed, at each time step $\Delta t = 20$ ms. Averaging the log band power over all electrodes of interest, yields the average log band power $z_t$. If the difference between the averaged baseline

[1] http://openvibe.inria.fr/

signal $b$, recorded in the relax phase and the incoming averaged log band power $z_t$ in the active phase, denoted by $d_t = z_t - b$ (signal-baseline difference), falls below a threshold $\theta$, a jump command $a_t^S$ will be sent to the game from the subject. Only in the active phase A, the signal-baseline difference $d_t$ is computed. The averaged baseline value $b$ is

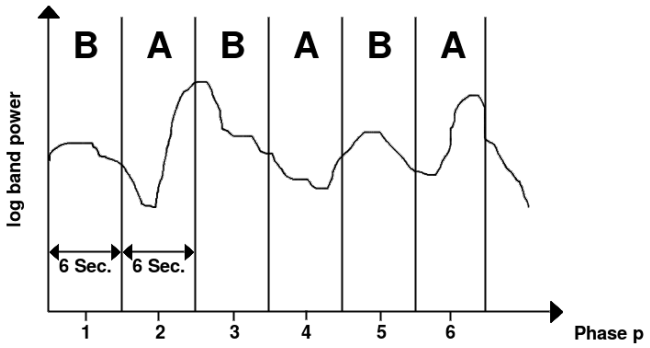



**Figure 4.2.:** A sketch to illustrate the different important phases in the game. For simplicity, the six second pause phase is ignored, because the signals in this phase are not relevant. B denotes baseline phases or relax phases. A denotes active phases. All phases take six seconds.

only computed in the baseline phase B. Each phase lasts six seconds. Lets assume that $N$ is the number of data points which arrive within the six seconds of a phase. The process of computing the signal-baseline difference $d_t$ and sending a command $a_t^S$ to the game can be formalized as follows.
First the averaged baseline value $b$ has to be computed in the baseline phase as

$$b = \frac{1}{N}\sum_t z_t\,, \quad \forall t \in \text{Baseline}\,. \tag{4.1}$$

At every time step of the active phase, the difference $d_t$ between the averaged baseline value $b$ from the previous baseline phase and the incoming averaged log band power value $z_t$ from the current active phase has to be calculated as

$$d_t = z_t - b\,, \quad \forall t \in \text{Active}\,. \tag{4.2}$$

To send a command $a_t^S$ to the game, the signal-baseline difference $d_t$ needs to be below a threshold $\theta$. Since it cannot be assumed that the moving average log band power is correct at the beginning of the experiment, an initial threshold is defined with a value of $-0.1$. The threshold adapts itself after each active trial and uses data from the last three active trials to calculate a running average over the aggregated, averaged log band power values. Using Equations (4.1) - (4.2), we can determine action $a_t^S$ as follows

$$a_t^S = \begin{cases} 1, & \text{if } d_t < \theta \\ 0, & \text{otherwise.} \end{cases}$$

If $a_t^S$ is 1, a jump command will be sent to the game from the subject, otherwise the game figure will remain on the ground.

#### Q-Learning

To assist the subject in jumping over the obstacle, an optimal reinforcement learning agent was trained with the Q-Learning algorithm. The agent is able to play the game optimally, i.e., through the learned policy the RL agent knows which is the right command to execute in every state of the game. At the beginning of the experiment, the agent has most of the control over the commands and executes them. As the experiment proceeds, the assistance of the agent decreases while the subject gets more own control over the actions.
The game takes place in a two dimensional world and thus the $x$ and the $y$ position of the player and the obstacle are the 2D world coordinates. The game figure only moves in $y$ direction and the obstacle only moves in $x$ direction. To reduce the number of possible states, we first limit the number of possible $x$ and $y$ values and further discretize the resulting value ranges.
A state $s_t$ consists of three values. The first value is the $y$ position of the player $y_{\text{player}} \in \mathbb{N}$, the second value is the $x$ position of the obstacle $x_{\text{obs}} \in \mathbb{N}$ and the third value $v_{\text{player}} \in \{0, 1\}$ describes if the obstacle is in a predefined vision range of the player, where $v_{\text{player}}$ is 1, if the obstacle is in the vision range of the player and 0 otherwise. Thus, a state $s_t$ is a

triple $(y_{\text{player}}, x_{\text{obs}}, v_{\text{player}})$. The possible set of actions **A** are a jump command, labeled as 1 or no command, labeled as 0. The discretized set of states **S** and the set of actions **A** can be formalized as

$$S = \{s_t \in (y_{\text{player}}, x_{\text{obs}}, v_{\text{player}})\},$$

$$A = \{a_t \in \{0, 1\}\}.$$

We had an overall number of 1276 states. We discretized the y-value range to 11 states and the x-value range to 58 states. The vision range $v_{\text{player}}$ only had two states. We trained the agent on 1000 episodes. The learning rate $\alpha$ was set to 0.2, the discount factor $\gamma$ was set to 0.7 and the epsilon greedy factor $\epsilon$ was set to 0.2 such that the agent takes random actions with a probability of 20% to explore the environment. Furthermore, the agent gets the same rewards as the subject, i.e., -1 point for a jump, 400 points for a successful episode, -50 for a collision with an obstacle.

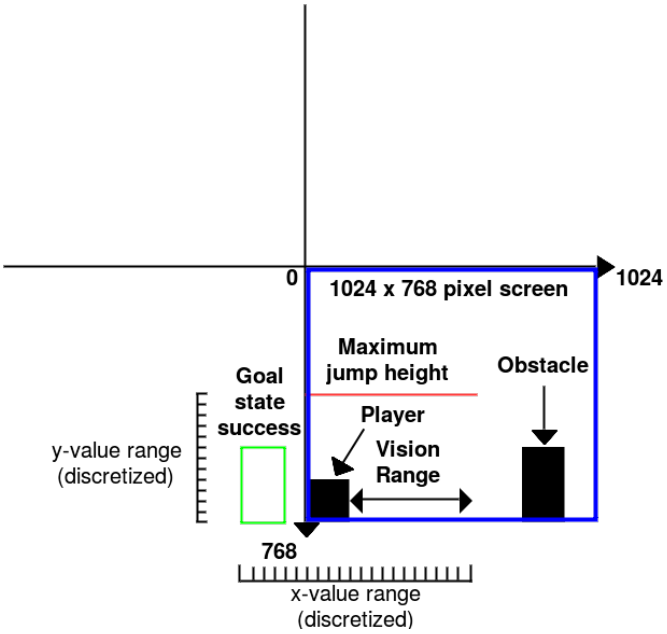



**Figure 4.3.:** An abstract sketch of the $1024 \times 768$ 2D world of the game. To decrease the number of possible states, we limit the $x$ and $y$ value ranges and discretize them to 11 states in height ($y_{\text{player}}$), 58 states in width $x_{\text{obs}}$. The player has a vision range and a maximum jump height. An episode is successful if the player manages to jump over the obstacle such that the obstacle arrives at the goal state, outside the pixel screen (blue rectangle). We have an overall number 1276 states.

#### Monte-Carlo sampling process

The decision if either the command of the subject or the RL agent is executed, is implemented through a Monte-Carlo sampling process. The current control signal of the agent is determined by a function which decreases with time (RL-control). The function is called the z-shaped membership function $f$ and can be formalized as

$$f(t; v, w) = \begin{cases} 1, & t \leq v \\ 1 - 2\left(\frac{t-v}{w-v}\right)^2, & v \leq t \leq \frac{v+w}{2} \\ 2\left(\frac{t-w}{w-v}\right)^2, & \frac{v+w}{2} \leq t \leq w \\ 0, & t \geq w, \end{cases}$$

where $t$ is the current time index in the game, $v$ is the time index where the z-shaped membership function $f$ starts to decrease and $w$ is the time index where $f$ converges to 0. The process works as follows. At every time step of the game, a random number $r$ will be generated. If this random number lies between 0 and the current value of the function $f$, then the action of the RL agent will be executed, otherwise the action of the subject will be executed, which can be formalized as

$$a_t = \begin{cases} a_t^{RL} & \text{if} \quad 0 \leq r \leq f(t; v, w) \\ a_t^{S} & \text{otherwise,} \end{cases}$$

where $a_t^{RL}$ is a command of the agent and $a_t^{S}$ is a command of the subject.

**Continuous feedback**
To provide the subject with continuous feedback, a scoring system was implemented. As previously mentioned, for every jump, the subject gets -1 point. The purpose of giving -1 point for a jump is to remove the bias of executing no action which occurs more often than jumps. Furthermore, as the subject should take the displayed points as feedback, the subject should stay on the ground as long as possible and jump over the obstacle at the right time, to loose as few points as possible. For a successful trial, the subject gets 400 points. For a collision with the obstacle, the subject gets -50 points. The subject should use the continuously displayed points as feedback and should try adapt to the slightly changing behavior of the game figure.

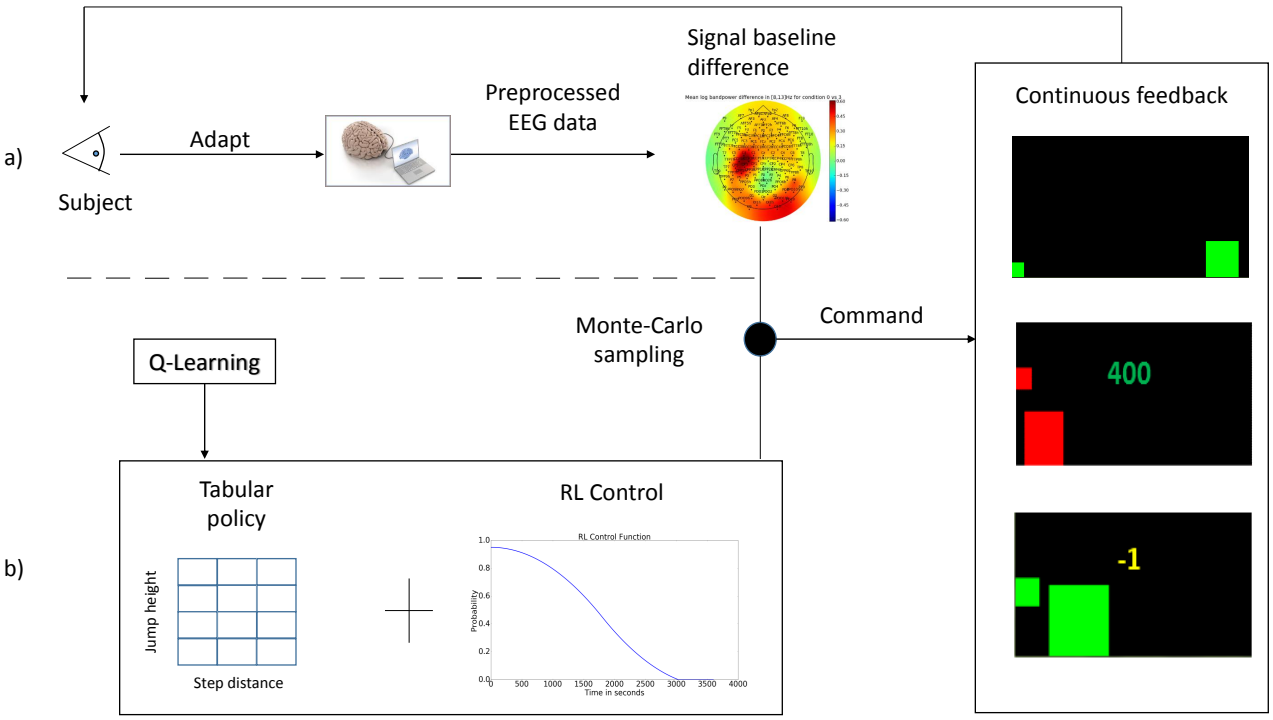



**Figure 4.4.:** In a), a standard BCI control scheme is shown. The subject imagines a movement with its left hand, right hand or feet. If the signal-basline difference, i.e., the activation of the respective brain region is strong enough and thus the difference passes a threshold, the subject is ready to send a jump command to the game. Otherwise the game figure stays on the ground, indicating no command. In b), a RL agent is trained to learn an optimal tabular policy. The state space is spanned over the jump height and step distance to the obstacle. Depending on the distance to the obstacle, the agent will send a jump command or no command to the game. The agent will only jump, if it is necessary. A selection policy (implemented through a Monte-Carlo sampling process), selects either the command generated by the subject or by the RL agent. A decision is made every 20 milliseconds.

## 4.2 Openvibe Implementation

To acquire, filter, process and classify brain signals in real time, Openvibe[1] is used. Openvibe is a open source software platform dedicated to designing, testing and using brain computer interfaces. Three different pipelines have been implemented. A training pipeline for the training phase, a classifier training pipeline to train the classifier with the data from the training phase and online decoding pipeline for the testing phase, where the subject gets confronted to new environments and the classifier should classify new incoming signals. The signal is sampled with 512 Hz.

**Subject training pipeline**
For each limb, we use a set of electrodes over the motor area, which is responsible for the movement of the limb. For the left hand, the electrodes C4, C2, C6, CCP4h, CCP6h, FCC4h, FCC6h were used, for the right hand, the electrodes C3,

[1] http://openvibe.inria.fr/

C5, C1, CCP5h, CCP3h, FCC5h, FCC3h were used and for the feet the electrodes FCz, FC1, FC2, FFC1h, FFC2h, FCC1h, FCC2h were used. To look at the electrode placement on the cap, see Figure (3.2b))
After the signal is pre-processed and the log band power is computed, see Subsection (4.1), the log band power is sent to a Python script. The Python script is responsible for maintaining and labeling the current game state in the signals, i.e., beginning of the experiment, beginning of a trial, relax phase, active phase, pause phase, end of a trial and end of the experiment. Furthermore, it will send a jump command to the game if the signal baseline difference falls below a threshold. To send a command to the game and ensure communication between the Openvibe pipeline and the Python script, a lab streaming layer (LSL) protocol is used.

**Classifier training pipeline**
The classifier training pipeline uses almost the same steps as the subject training pipeline to pre-process the signals and to compute the log bandpower. The only difference is that, we used a 100 ms sliding window which is applied over three second epochs to train the classifiers. The number of data points would be to high with a 20 ms sliding window, which would result in a long training time of the classifiers. The log bandpower of each electrode is used as a part of the feature vector. Since for the experiment only two of three limbs were used, a 14 dimensional feature vector consisting of the log bandpower of each electrode is used for the classifier. Since brain signals can change during the training, only the training data of the last 30 minutes is used to train the classifier.

**Online decoding pipeline**
The pre-processing of the data and the computation of the log bandpower is the same as for the subject training pipeline. Here, a 20 ms sliding window is applied over three second epochs again to match the 50 Hz of the training phase. The trained classifier tries to classify incoming EEG signals. Depending on the current game state and the classification of the signal, a jump command will be sent to the game, e.g., if the game state is that the subject should think about the left hand and the classifier classifies the left hand, then a jump command will be sent to the game. In case of a wrong classification, the game figure remains on the ground.

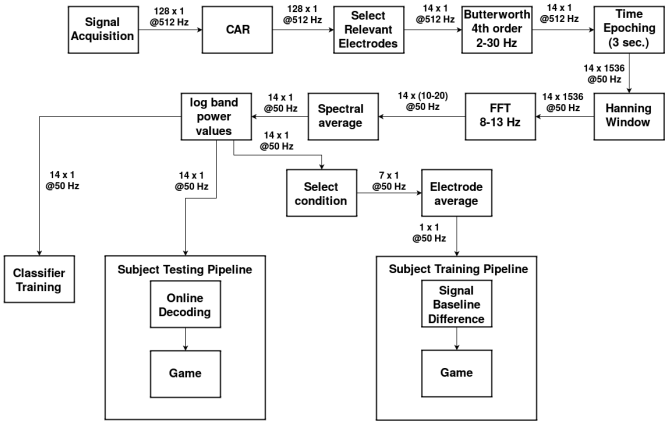



**Figure 4.5.:** This sketch illustrates the flow of the used Openvibe pipelines. The signal goes through several pre-processing steps, before the log band power of the signal is computed. The subject training pipeline uses the log band power features to calculate the signal baseline difference and send a command to the game in the training phase. A classifier is trained with the log band power features between the training and the testing phase. The subject testing pipeline uses the trained classifier to decode incoming signals and send command to the game in the testing phase.

# 5 Experiments and Results

In this chapter, we present the setup of the experiment and how it was conducted. Furthermore, we present the results of the experiments.

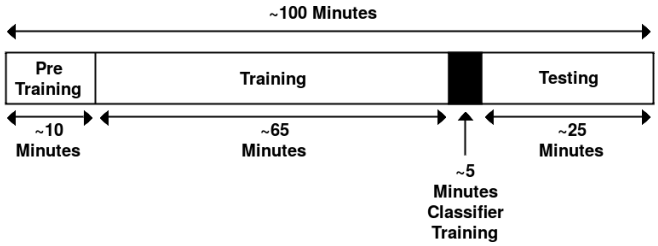



**Figure 5.1.:** This sketch illustrates the different stages of the experiment and the duration of each stage. The whole experiment took about 100 minutes.

## 5.1 Experimental Setting

During the experiments, a 128 electrode actiCHamp system from BrainProducts[1] was used to acquire brain signals. Each subject sat in a comfortable chair in front of a computer screen where the game was displayed. The impedance of each electrode was gelled to under 20 kOhm. Two groups were evaluated. Each group consisted of ten subjects. All of the subjects never participated in a BCI experiment before. One group did the experiment with the RL agent but is not aware of the RL agent and one group did the experiment without the RL agent.
To tune the parameters for the experiment, that are listed in Table (A.1 - A.5), such that we get the best possible performance out of the subject and the system, we did some pre-tests on four subjects. All the subjects who agreed to take part in the pre-tests already worked with a BCI or took part in other BCI experiments. All participants gave feedback after the experiment. The feedback was used to improve the system. The parameters to tune can be grouped into 5 groups. The experimental settings, Openvibe pipeline settings, game settings, reinforcement learning settings and classifier settings. The parameters can be looked up in the parameter table in the appendix, with an explanation for each parameter and the values we used during the pre-tests. Most parameters for the Openvibe pipeline were chosen based on the BCI literature [29]. Parameters for the experimental settings, like the duration of a condition for example, were chosen due to the experience from previous similar experiments done with a BCI.

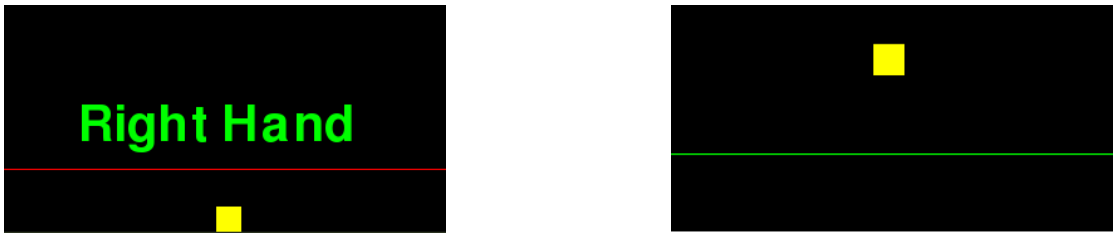
#### Pre-Training phase

The first phase of the experiment is the pre-training phase. Here, the subject has the possibility to get used to the system and using motor imagery to make the figure jump. This part of the experiment has no approaching obstacles. Just a game figure with a horizontal line is displayed, see Figure (5.2a)). In the active phase of every trial, the subject gets displayed which limb it should imagine to move to make the figure jump over the horizontal line and to keep it above the line. As the figure gets above the line, the line turns from red to green, see Figure (5.2b))
Each trial consists of a relax, active and pause phase. Each phase takes six seconds. This part of the experiment has one run. Each run consists of eight trials per condition. The conditions are the left hand, right hand and feet. After this phase is completed, the subject chooses two of three conditions which worked best in this phase. These conditions will be used in the training phase. Most of the time, the left and the right hand were chosen by the subjects. This phase took about ten minutes.

#### Training phase

In the actual training phase, every group did eight runs where one run consists of 20 trials. Each trial consists of a six second relax phase where the baseline is recorded, a six second active phase, where the subject should imagine a movement with one of its limbs, and a six second pause phase, where the subject can move if it needs to. Between runs, the subject has a pause of one minute in the first four runs and a pause of two minutes in the remaining four runs. In the active phase of each trial, an obstacle approaches the game figure where the game figure needs to jump over the obstacle. This phase took about 65 minutes.

[1] http://www.brainproducts.com/




(a) (b)

**Figure 5.2.:** In the pre-training phase, the subject gets the possibility to get used to motor imagery and the BCI system. In the pre-training phase there are no approaching obstacles. The subject needs to use imagined movements to get the figure over a red horizontal line, shown in a), which turns green, as soon as the game figure crosses the line, shown in b).

**Testing phase**

In this part of the experiment, a linear support vector machine was trained with the data from the training phase. To avoid a bias of the classifier on one condition, we only took the last 30 minutes of data to train the classifier. Both groups played without the RL agent and were confronted to three new obstacles approaching the game figure. The testing phase consisted of three runs, one run for each obstacle. A run had 20 trials, where a trial consisted of a six seconds active phase followed by a six second pause phase. We dropped the relax phase, because the classifier should correctly classify one class label, the relaxing condition of the subject from the incoming data to keep the game figure on the ground. We again used the two conditions the subject chose for the training phase. Each condition was displayed ten times per run. This phase took about 25 minutes.

## 5.2 Impact of RL Guidance on Learning Performance

The effect of the RL agent was evaluated in the training phase. To evaluate the performance of both groups, we logged every point for each subject during the experiment and calculated the points a subject reached in a run. We averaged the points of a run over all subjects for each group and compared the averaged run points of both groups against each other. Furthermore, we compared the taken actions from the RL agent and the subjects in a run against the success rate of the subjects from the RL group to see, how successful the subjects were on average while the help of the RL agent decreased with time. We also evaluated a trend plot for both groups. The trend plot shows if one of the groups was able to modulate their log band power during the experiments. For the trend plot, we calculated the average log band power baseline and active values for each trial and fitted a linear function to the plot to see, if there is a trend for each of the groups.

Looking at the average points for each run in Figure (5.3), the RL group seems to outperform the control group. Especially last run of the RL group, where the RL group plays without help of the RL agent, outperforms every run of the control group. If we look at the taken actions versus the average success rate from the RL group in Figure (5.4b)), we see that from the fifth run on, where the subjects took more actions than the RL agent, the average success rates are better than the success rate from the fourth run. Comparing the trend plots of both groups in Figure (5.5a)) and Figure (5.5b)), we can see that the RL group was able to modulate their log band power during the experiments, while the control group was barely able to modulate their log band power.

If we recap all results, increasing self-control seems to have an impact on the subjects from the RL group. Motivation and learning seem to benefit from the RL-agent, compared to the control group.

## 5.3 Generalization to new Environments

To evaluate if the subject is able to adapt to new situations, three new obstacles are presented to the subject. A faster, a wider and a higher obstacle is introduced. The data gathered in the training phase is used to train a linear support vector machine for three classes, the baseline class and two classes for the conditions of the motor imagery paradigm, see Subsection (3.1) on SVMs. The feature vector is a 14 dimensional vector consisting of log band power values for each electrode. Since brain signals are non-stationary and a classifier can become biased on one condition due to this non-stationarity, the classifier is trained with the most recent data, only using the data of the last 30 minutes. Furthermore, a ten fold cross validation is applied to the data and a one versus one strategy is used. The cost factor $C$ was set to seven. The cost factor determines the cost for a misclassification. A low cost factor has the risk to underfit the data, while a high cost factor has the risk to overfit the data. We compared the percentage of correctly executed actions in each trial and

looked if the trial was successful. To do that, we took the best subject for each group and compared the executed actions of the subject to the ground truth, i.e, the optimal actions.
Looking at the results in Figure (5.6 - 5.8), the subjects of both groups seemed not to be able to reliably control their actions. For both subjects there is no pattern that indicates that they were able to adapt to the classifier over time. Sometimes even a high percentage of correctly executed actions did not lead to success, whereas a low percentage of correctly executed action lead to success, indicating that the figure was jumping all the time. Possible reasons for that behavior will be discussed in the next chapter.

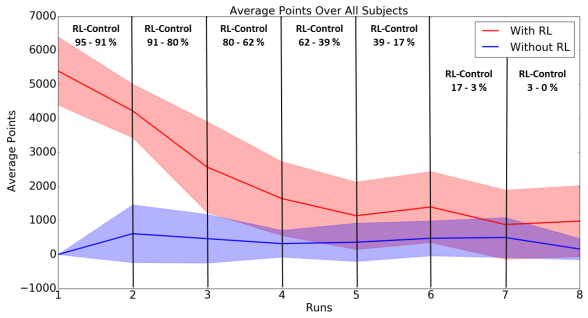
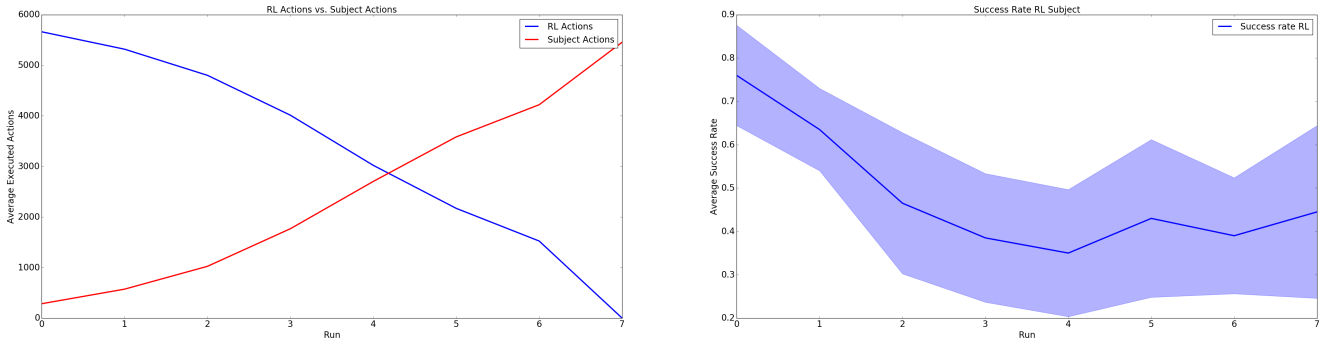


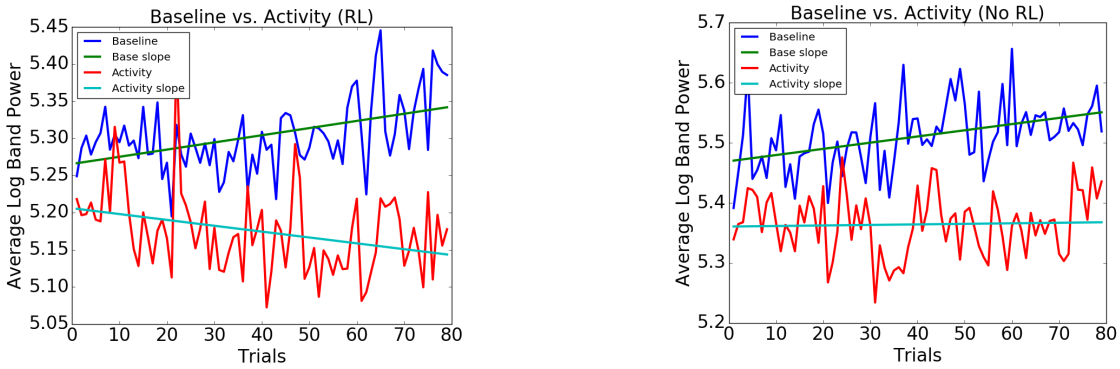

**Figure 5.3.:** Here, we show the average points over all subjects for each group was calculated. The red curve denotes the average points, collected in each run for the RL group. The blue curve denotes the average points, collected in each run for the control group. The percentage values show, how much the RL assistance drops between runs.




**(a)** Executed actions of RL agent and subject

**(b)** Success rate of subject

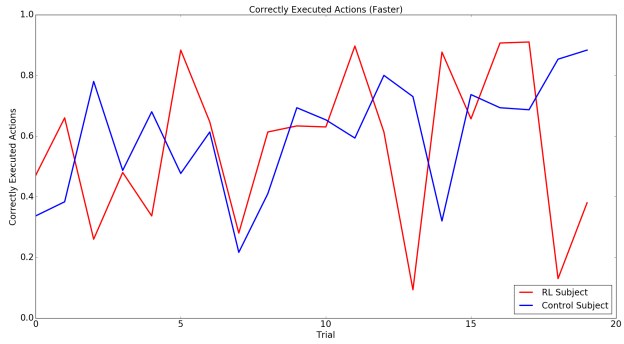
**Figure 5.4.:** In a), we show the average of executed actions for each run. The blue curve denotes the actions of the RL agent, while the red curve denotes the actions of the subject. In b), we show how the average success rate develops based on the decreasing impact of the RL agent.



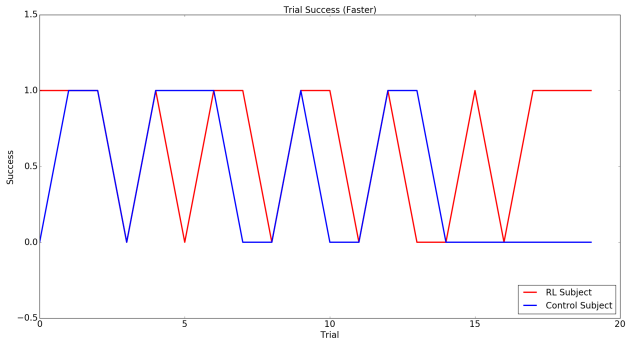

**(a)** Signal versus baseline RL group

**(b)** Signal versus baseline control group

**Figure 5.5.:** In a), we compared the average log band power of all baseline trials (blue curve) against the average log band power of all active trials (red curve) for the RL group to see, if the subjects were able to modulate their signals. In b), we compared the average log band power of all baseline trials (blue curve) against the average log band power of all active trial (red curve) for the control group. For both plots, we fitted a linear regression function to the curves to display the trends.

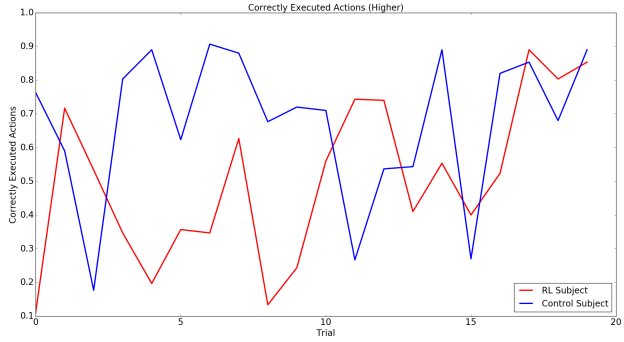



**(a)** Correctly executed actions

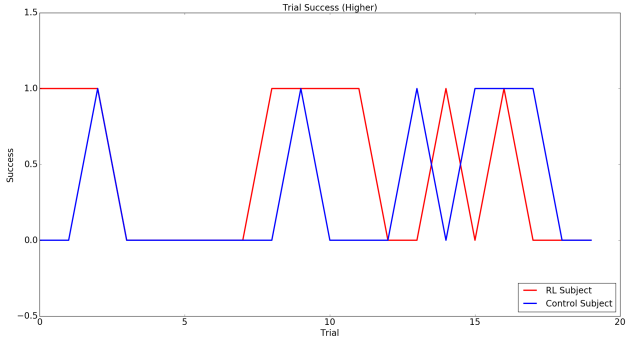



**(b)** Trial success

**Figure 5.6.:** In a), we show the average of correctly executed actions (faster obstacle) in each trial for the faster obstacle. We took the best subject from each group. In b), we show if a trial was successful or not for both subjects. One means that the trial was successful and zero means that a collision occurred.

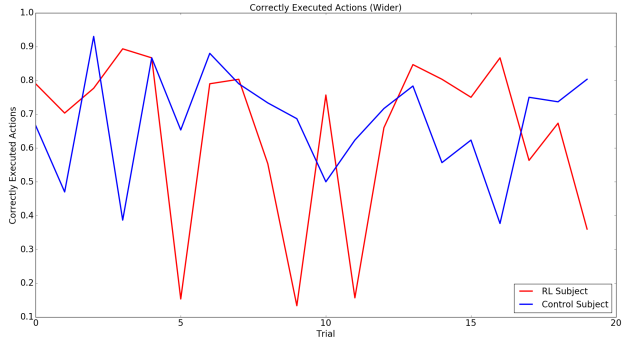



**(a)** Correctly executed actions

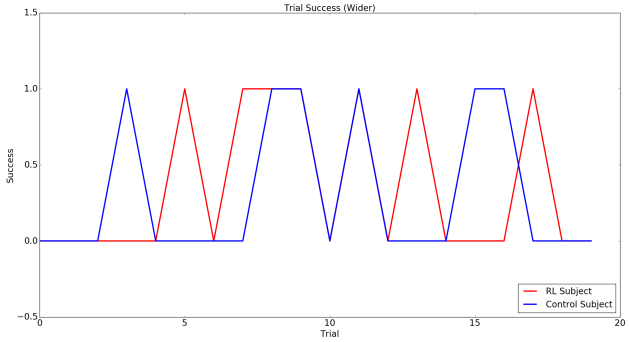



**(b)** Trial success

**Figure 5.7.:** In a), we show the average of correctly executed actions (higher obstacle) in each trial for the faster obstacle. We took the best subject from each group. In b), we show if a trial was successful or not for both subjects. One means that the trial was successful and zero means that a collision occurred.




**(a)** Correctly executed actions




**(b)** Trial success

**Figure 5.8.:** In a), we show the average of correctly executed actions (wider obstacle) in each trial for the faster obstacle. We took the best subject from each group. In b), we show if a trial was successful or not for both subjects. One means that the trial was successful and zero means that a collision occurred.

# 6 Conclusion and Future Work

## 6.1 Conclusion

In this work, we examined if subjects controlling a brain computer interface (BCI) system were able to learn faster or better policies with reinforcement learning (RL) assistance compared to classical training. For this purpose, we implemented a game, where the subject needed to control a game figure with its thoughts to try to jump over approaching obstacles. Here, the goal was not only to jump over obstacles, but rather to develop an optimal control policy for the game figure, making the figure jump or keeping it on the ground. A trained RL agent, who was able to play the game optimally assisted the subject in controlling the game figure by executing optimal actions. The RL assistance continuously decreased during training and was deactivated in the testing phase. We used a scoring system, where rewards and averaged scores over all subjects in each group are compared.
Looking at the results of the training, the RL group could outperform the control group in each run. However, as the RL group played with RL assistance, it rather makes sense to look at the average performance of the last three runs, where the subject took more actions than the RL agent. In the last run, the agent was completely deactivated. Interestingly, the RL group was able to increase its performance again after the fifth run, see Figure (5.4b)). If we compare the average performance of the last run from the RL group with the average performances of each run from the control group, the RL group was able to outperform the control group. The average performance of the control group almost dropped to zero in the last run, while the average performance of the RL group slightly increased in the last run compared to the previous run. Furthermore, the average log band power trend plot of both groups shows that the RL group was able to modulate their average log band power over time, while the control group was barely able to modulate their average log band power, see Figure (5.5a)) and (5.5b)).
We also evaluated if the subjects who trained with RL assistance were also to adapt to new environments faster than subjects who were not trained with the RL assistance. We confronted the subjects to faster, higher or wider obstacles to investigate if they were able to adapt their brain signals to new situations. For this purpose, we trained a classifier with the data acquired in the training phase. Assuming that brain signals also change during the training, we only used the most recent data (4 runs) to train the classifier. The classifier classified incoming data from the BCI system in real time. We only performed one run with 20 trials per obstacle as the subjects already worked with the BCI system for one hour and got tired.
The overall performances of both groups in the testing phase was not good. Both groups were not able to control the game figure. If we compare the ground truth, i.e., the actions that should be executed in each time step to reach optimal performance, and the executed actions of the best subject for each group, there is no visible pattern that indicates a continuous increase in correctly executed actions, see Figure (5.6a)) - (5.8b)). Potential solutions are discussed next.

## 6.2 Discussion

Since controlling a BCI requires a high level of concentration, the subjects ideally need to be in a good mental shape on the day of the experiment. Subjects being tired or having problems to concentrate may perform worse on average than subjects that are mentally in a good shape on the day of the experiment. Since we observed a large variance in the subjects performance, many more subjects need to be evaluated to make a significant statement about increasing self-control and learning to control a BCI.
In [8], the authors speculated to consider the skill level of a subject, when performing a BCI training scenario with biased feedback. We could also choose a less strict RL control function, giving a skilled subject more self-control over the figure from the beginning than a less skilled subject. Choosing a strict RL-control function for a long period, could make the subject think that motor imagery works even though their log band power values are bad. As the help of the RL agent decreases, the subject could get confused asking itself why motor imagery does not work anymore.
To successfully jump over the obstacles, the subject needs to solely concentrate on the movement it is imagining. Some subjects found it difficult to concentrate on imagining a movement and at the same time watching the obstacle approaching the game figure and trying to jump over it. The closer the obstacle got to the game figure, the more difficult it was for some subjects to concentrate on the movement. Mainly because the subjects got nervous with the concern to collide with the obstacle. The result of it was that the signal got unstable and went above the threshold again with the consequence that the game figure action was wrong. Thus, when designing an experiment, the subject should have more time to solve

the task, because the slightest deviation of the concentration could lead to failure.
A possibility for the bad performance in the testing phase could be amount of data points, we trained the classifier with. We needed to throw away some data points to prevent biasing of the classifier on one condition. However, to learn a good model of the data, the classifier needs enough data points. Gathering more data points would increase the training time for the subject and we still would have the problem of the non-stationarity of the data. Furthermore, we only used five minutes per obstacle, considering that the subject already trained over an hour. This might be too little time to adapt to new situations.

## 6.3 Future Work

**Incremental learning.** In this thesis, we used two commands to train the subjects. However, to learn to control a BCI, it would be more suitable to focus first on a simple command. After successfully learning one command, the subject could apply the same learning patterns on other commands, trying to successfully learn the other commands. In future work, we are investigating such transfer learning BCI techniques.

**Skill level dependent testing.** During the experiment, we chose a very strict selection policy, starting at 95 % which means that the agent executed the actions most of the time. Choosing a different selection policy, based on the skill of the subject could improve the results. The question here is, if the subjects would perform even better with a less strict selection policy, already giving more control to the subject in the beginning of the experiment. The authors in [8] speculated that biased feedback should take the current skill level of the subject into account.

**Duration of a task.** Some subjects had difficulties concentrating on imagining a movement while observing the obstacle approaching the game figure. The nearer the obstacle got, the more nervous the subject was, which resulted in a collision with the obstacle. Thus, it should be considered to implement a task where the subject is not under time pressure, giving it enough time to solve a task. In future experiments, these parameters will be carefully evaluated.

**Co-adaptive learning.** Instead of using three new obstacles, a more suitable step would be to see, if the subject is able to play with the classifier on the obstacle, we used in the training phase. Then we could use the 25 minutes for the standard obstacle, where the subject would have more time to adapt to the classifiers output first. Additionally, we could use an online learning approach, such that the classifier is trained on new incoming singals. The procedure that the subject adapts to the classifiers output and the classifier adapts to new EEG signals of the subject is called co-adaptive learning [11]. After the subject is able to play with the standard obstacle, we can introduce new obstacles to see, if the subject is able to adapt to new situations.

**Training with low cost BCIs.** Using a high density EEG also means spending a lot of time in gelling each electrode to reduce the signal to noise ratio. This could also affect the subject in a negative way. Therefore, using a low cost BCI or a EEG with less electrodes can save time in preparing the experiment. Of course the signals will not have the same quality as with a high density EEG but it would be worth trying to see if our presented approach works with a low cost BCI, like the Emotiv[1] BCI system for example.

**BCI Features.** In this work, we used simple bandpower feature spaces which only consider changes in the power of the frequency bands. More complex feature spaces like Riemann covariance [30] feature space also consider spatial correlations between electrodes which could improve classifying results for the classifiers, resulting in a better performance of the classifier and the subject.

**Multi-task transfer learning.** One of the biggest problems in BCI research is the non-stationarity of brain signals. This non-stationarity makes it difficult for a classifier to find reliable patterns in the signals, resulting in bad classifying performances. Recently, a method was developed [31, 32] which calculates a prior over the data to capture common structure in data. This makes it possible to transfer common knowledge over sessions or even over subjects. In combination with the RL agent or other feedback approaches this could improve learning to control a BCI and reduce training time for the subject, since we could use a pre-trained universal classifier which just needs to be adapted to the new subject.

[1] https://www.emotiv.com/

# Bibliography


[1] E. Kandel, J. Schwartz, T. Jesell, S. Siegelbaum, and A.J.Hudspeth, *Principles of Neural Science*. McGraw Hill Professional, 2013.

[2] BrownUniversity, "BCI Robotarm." https://news.brown.edu/files/styles/horizontal/public/article_images/DrinkingMoment.jpg?itok=UPhMcx4R, 2017. [Online; accessed 15-January-2017].

[3] YouTubeImage, "BCI Exoskeleton." https://i.ytimg.com/vi/Jcode0OLw7U/hqdefault.jpg, 2017. [Online; accessed 15-January-2017].

[4] BerlinBrainComputerInterface, "BCI Brainpong." http://www.bbci.de/images/bbci_brainpong_s.png, 2017. [Online; accessed 15-January-2017].

[5] S. Ogawa, T. Lee, A. Kay, and D. Tank, "Brain magentic resonance imaging with contrast dependent on blood oxygenation," *Proc Natl Acad Sci U.S.A.*, vol. 87, no. 90, pp. 9868–9872, 1990.

[6] P. Shenoy, M. Krauledat, B. Blankertz, R. P. N. Rao, and K.-R. Müller, "Towards adaptive classification for BCI.," *Journal of neural engineering*, vol. 3, no. 1, pp. R13–R23, 2006.

[7] R. Leeb, F. Lee, C. Keinrath, R. Scherer, H. Bischof, and G. Pfurtscheller, "Brain computer communication: Motivation, aim, and impact of exploring a virtual apartment," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 15, pp. 473–482, Dec 2007.

[8] A. Barbero and M. Grosse-Wentrup, "Biased feedback in brain-computer interfaces.," *Journal of neuroengineering and rehabilitation*, vol. 7, p. 34, 2010.

[9] H.-j. Hwang, K. Kwon, and C.-h. Im, "Neurofeedback-based motor imagery training for brain – computer interface ( BCI )," vol. 179, pp. 150–156, 2009.

[10] M. Gomez-Rodriguez, J. Peters, J. Hill, B. Schölkopf, A. Gharabaghi, and M. Grosse-Wentrup, "Closing the sensorimotor loop: Haptic feedback facilitates decoding of arm movement imagery," *Journal of Neural Engineering*, vol. 8, no. 3, 2011.

[11] C. Vidaurre and B. Blankertz, "Towards a cure for BCI illiteracy," *Brain Topography*, vol. 23, no. 2, pp. 194–198, 2010.

[12] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[13] S. I. H. Tillery, D. M. Taylor, and A. B. Schwartz, "Training in cortical control of neuroprosthetic devices improves signal extraction from small neuronal ensembles.," 2003.

[14] NHSChoice, "Positron Emission Tomography." http://www.nhs.uk/Conditions/PET-scan/PublishingImages/M410302-PET_scanner_342x198.jpg, 2016. [Online; accessed 30-December-2016].

[15] Technomaly, "Electrocorticography." http://www.technomaly.com/wp-content/uploads/2010/02/ECoG.jpg, 2016. [Online; accessed 30-December-2016].

[16] TrueImpact, "Functional magnetic resonance imaging." http://www.trueimpact.ca/wp-content/uploads/2013/03/GE-fmri-machine.jpg, 2016. [Online; accessed 30-December-2016].

[17] Biopac, "Functional near-infrared spectroscopy." https://www.biopac.com/wp-content/uploads/fnir200.jpg, 2016. [Online; accessed 30-December-2016].

[18] UniversityOfWashington, "Magentoencephalography." http://ilabs.washington.edu/sites/default/files/MEG_art.jpg, 2016. [Online; accessed 30-December-2016].

[19] J. Carmena, M. Lebedev, R. Crist, J. O'Doherty, D. Santucci, D. Dimitrov, P. Patil, C. Henriquez, and M. Nicolelis, "Learning to control a brain machine interface for reaching and grasping by primates," *PLoS Biology 1(2): e42*, 2003.

[20] Athena-Minerva-Team-TU-Darmstadt, "BCI Pipeline." http://www.cybathlon.informatik.tu-darmstadt.de/uploads/Research/Brain-ComputerInterfacing/bci_simplified.png, 2017. [Online; accessed 23-January-2017].

[21] R. P.N.Nao, *Brain-Computer Interfacing: An Introduction*. Cambridge University Press.

[22] F. Lotte, M. Congedo, A. Lécuyer, F. Lamarche, and B. Arnaldi, "A review of classification algrithms for eeg-based brain-computer interfaces.," *Journal of Neural Engineering, IOP Publishing, 4*, pp. 24–, 2007.

[23] D. McFarland, L. McCane, S. David, and J. Wolpaw, "Spatial filter selection for eeg-based communication," *Electroencephalographic Clinical Neurophysiology 103(3):386-394*, 1997.

[24] F. Lotte, *A Tutorial on EEG Signal Processing Techniques for Mental State Recognition in Brain-Computer Interfaces. Eduardo Reck Miranda; Julien Castet. Guide to Brain-Computer Music Interfacing*. Springer Book, 2014.

[25] C. M. Bishop, *Pattern Recognition and Machine Learning*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[26] C. Cortes and V. Vapnik, "Support-vector networks," in *Machine Learning*, pp. 273–297, 1995.

[27] D. Sharma, *Combining Reinforcement Learning and Feature Extraction*. Bachelor's thesis, "2012".

[28] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex fourier series," *Math. Comput. 19*, 1965.

[29] J. Wolpaw and E. Wolpaw, *Brain Computer Interfaces: principles and practice*. Oxfort University Press, 2012.

[30] A. Barachant, S. Bonnet, M. Congedo, and C. Jutten, "Classification of covariance matrices using a Riemannian-based kernel for BCI applications," *Neurocomputing*, vol. 112, pp. 172–178, July 2013.

[31] K.-H. Fiebig, V. Jayaram, J. Peters, and M. Grosse-Wentrup, "Multi-task logistic regression in brain-computer interfaces," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC 2016)*, IEEE, 2016.

[32] V. Jayaram, M. Alamgir, Y. Altun, B. Schölkopf, and M. Grosse-Wentrup, "Transfer learning in brain-computer interfaces," *IEEE Computational Intelligence Magazine*, vol. 11, no. 1, pp. 20–31, 2016.

# A Appendix

## A.1 Parameter Tables

| Parameter | Tested value(s) | Comments |
|---|---|---|
| Conditions | **Left hand, right hand, feet** | Limbs, the subject should imagine a movement with |
| Number of runs | **8** | A run consist of $n$ trials |
| Number of trials per condition | **Left hand: 10,**<br>**Right hand: 10,**<br>**Feet: 10** | Number of times, an obstacle appears in a run. A trial consists of a relax, active and pause phase. The obstacle appears in the active phase. |
| Relax duration | **6 seconds** | The time, the relax phase is active. |
| Active duration | **6 seconds** | The time, a subject has to jump over the obstacle. |
| Pause duration | **6 seconds** | The time, a subject has the possibilty to move. The signals from this phase are ignored. |
| Pause between runs | **1 minute in the first 4 runs and 2 minutes in the last 4 runs.** | The time a subject has the possibility to take a break from the experiment. |
| Transition phase: Relax → Active | Fixed: 5 seconds,<br>**Random interval between 0.1 and 3 seconds.** | The time from where the pause phase ends and the relax phase starts. A random number between 0.1 and 3 seconds will be generated such the subject cannot prepare for the following phase. |
| Transition phase: Pause → Relax | Fixed: 5 seconds,<br>**Random interval between 0.1 and 3 seconds** | The time from where the pause phase ends and the relax phase starts. A random number between 0.1 and 3 seconds will be generated such the subject cannot prepare for the following phase. |
| Adaptive threshold for the signal-baseline difference | 0,<br>**Initial value: -0.1 (average log band power value)** | The threshold adapts itself to the average log band power value of the last three activity trials, approximately 900 data points. A data point arrives every 20 ms. |

**Table A.1.: Experimental settings parameters.** These parameters define the experimental settings. Parameters like relax duration, activity duration and pause duration were chosen due to previous experience from other, similar experiments with BCIs. The transistion phases were chosen to be random, because the subjects should not be prepared for the next phase of the experiment. Parameters like number of runs and pause between runs were critical for the duration of the experiment. The bold values are the finally used values for the experimental settings.

| Parameter | Tested value(s) | Comments |
|---|---|---|
| Electrodes for the left hand | {C4}, **{C4, C2, C6, CPP4h, CPP6h, FCC4h, FCC6h}** | The average log band power value of all electrodes will be calculated for every incoming data point. |
| Electrodes for the right hand | {C3}, **{C3, C5, C1, CCP5h, CCP3h, FCC5h, FCC3h}** | The average log band power value of all electrodes will be calculated for every incoming data point. |
| Electrodes for the feet | {FCz}, **{FCz, FC1, FC2, FFC1h, FFC2h, FCC1h, FCC2h}** | The average log band power value of all electrodes will be calculated for every incoming data point. |
| Frequency bands | **Alpha range: 8-13 Hz** | Frequency band were the change of power is expected if a movement is imagined. |
| Temporal Filter | **Butterworth filter of 4th order to focus on 2-30 Hz** | Filter to reject unwanted frequencies and focus on important frequencies in the time domain. |
| Temporal epoching with sliding window | 3 second epochs with a 100 ms sliding window, **3 seconds epochs with a 20 ms sliding window** | 3 second data chunks were used with a 20 ms sliding window such that the data arrives with 50 Hz. |
| Windowing function | **Hanning window** | Smooths the signal |
| Spatial filter | **Common average reference** | Subtracts common activity over all electrodes from every electrode. |
| Stimulation offset | **0.5 seconds** | Ignores the first 0.5 seconds after the stimulation appeared (for classification.) |

**Table A.2.: Openvibe pipeline settings.** These parameters were important for the signal processing part of the experiment. They were chosen due to the fact that we used motor imagery as the paradigm to train the subject with. The sliding window defined the frequency of arriving data points to the game. The bold values are the finally used values for the Openvibe pipeline.

| Parameter | Tested value(s) | Comments |
|---|---|---|
| World dimensions | **Width: 1024 pixels**<br>**Height: 768 pixels** | Properties of the game. |
| Number of obstacles | **3** | Every obstacle is mapped to one condition. |
| Obstacle width, height and speed (training) | Speed: 5 pixels,<br>**Width: 50 pixels**<br>**Height: 50 pixels**<br>**Speed: 3 pixels** | Properties of the obstacle. |
| Faster obstacle: width, height, speed | **Width: 50 pixels**<br>**Height: 50 pixels**<br>**Speed: 5 pixels** | Properties of the faster obstacle. |
| Higher obstacle: width, height, speed | **Width: 50 pixels**<br>**Height: 70 pixels**<br>**Speed: 3 pixels** | Properties of the higher obstacle. |
| Wider obstacle: width, height, speed | **Width: 71 pixels**<br>**Height: 50 pixels**<br>**Speed: 3 pixels** | Properties of the wider obstacle. |
| Player: Jump power | 3 pixels per frame,<br>5 pixels per frame,<br>**7 pixels per frame** | Jump power of the player. |
| Player: Max jump height | No limit,<br>**Until pixel 698** | The player starts from the bottom, i.e., pixel 768 and can maximally jump to pixel 698 |
| Player: Vision range | 10 pixels,<br>20 pixels,<br>50 pixels,<br>**100 pixels** | The vision range of the player. This is important for the reinforcement learning algorithm to learn the game optimally. If the obstacle is in the vision range the state vector of the algorithm changes. |

**Table A.3.: Game setting parameters.** These parameters were used for the presentation of the game to the subject. The speed was the most task relevant parameter for the subject, because the speed determines the time, a subject has, to jump over the obstacle. Parameters like the jump power, max jump height, speed and vision range were relevant for the size of the state space for the RL agent. The bold values are the finally used values for the game settings.

| Parameter | Tested value(s) | Comments |
|---|---|---|
| Number of states | **1276** | Total number of states in the 2D world. The states are discrete. |
| Number of actions | **2** | Either a jump command, labeled as 1 or no command, labeled as 0. |
| Number of episodes | 500, 2000, 3000, 10000, **1000** | Number of times, the agent tries to solve the task. |
| Learning rate | 0.1, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.15, 0.25, 0.35, **0.2** | Determines the importance of new information. |
| Discount factor | 0, 0.1, 0.2, **0.7** | Determines the importance of future rewards. |
| Epsilon greedy value | 0.3, 0.5, 0.7, **0.2** | There is a possibility of 20% that the chosen action is not optimal. This is needed to explore the environment. |
| Reward for success | 50 points, 100 points, **400 points** | Reward, the agent gets for a successful episode. |
| Reward for collision | -100 points, **-50 points** | Reward, the agent gets for a failed episode. |
| Reward for jump | 0 points, **-1 point** | Reward, the agent gets for a jump in an episode. |

**Table A.4.: Reinforcement learning parameters:** These parameters were used to train the reinforcement learning agent on the task to assist the subjects in solving the task. The bold values are the finally used values for the reinforcement learning algorithm.

| Parameter | Tested value(s) | Comments |
|---|---|---|
| Classifier | Linear discriminant analysis, **Support vector machine** | Classifier to decode signals. |
| Multiclass strategy | One vs. All **One vs. One** | Three classifiers are learned because we use three classes, i.e., 0 vs. 1, 0 vs. 2 and 1 vs. 2. |
| Kernel type | Radial basis function, **Linear** | We used a linear decision boundary to classify the data. |
| C | 1, 2, 3, 4, 5, 6, 8, 10, 100, 1000, **7** | Cost factor. A small value has a low cost of for misclassification, while a large value has a high cost for misclassification. |

**Table A.5.: Classifier parameters:** These parameters were used to train the classifier. The bold values are the finally used values for the classifier

## A.2 Ethical Approval

Die Ethikkommission

TECHNISCHE UNIVERSITÄT DARMSTADT

# Checkliste
# zur ethischen Bedenklichkeit von Forschungsvorhaben

1) Sind im Rahmen Ihres Projekts Gefahrstoffe (Chemikalien, biologische Agenzien, radioaktive Materialien) im Einsatz – ggf. auch innerhalb gesetzlicher Grenzen oder werden solche wissentlich oder potentiell erzeugt? Ist dies für alle Beteiligten transparent und werden maximale Schutzmaßnahmen getroffen?

Es sind keine Gefahrenstoffe im Einsatz. Im Rahmen des Experiments mit dem EEG muss dem Probanden/der Probandin leitendes Gel zwischen Kopfhaut und Elektrode mit stumpfen Spritzen gespritzt werden, damit die EEG Signale besser gemessen werden können. Das Gel lässt sich mit Wasser und Seife wieder entfernen.

2) Sind mit der Durchführung Ihres Forschungsvorhabens für Beteiligte körperliche Gefahren (z.B. durch Laserstrahlen, Lärm, starke sportliche Belastung) verbunden?

Nein. Die Kopfhaut muss jedoch mit Stumpfen spritzen aufgeraut werden. Dabei kann es eventuell zu Hautrötungen kommen, die nach einigen Stunden bis hin zu wenigen Tagen wieder abklingen.

3) Bei Forschungsvorhaben mit Versuchen an oder mit Menschen: Wieviele Probanden sind vorgesehen? Welche Dauer haben die Untersuchungen für die einzelnen Probanden und welchen Belastungen sind sie dabei ausgesetzt? Werden die Probanden besonderer Merkmale wegen ausgesucht?

Für die Experimente sind bis zu 20 Probanden vorgesehen. Zusätzlich werden im Laufe von Vorstudien 3-4 Probanden eingeladen um etwaige Fehler in der Software zu beseitigen. Die Versuche werden pro Proband ca. 1 Stunde inklusive Pausen andauern. Dabei wird der Proband instruiert an eine Bewegung mit Hand oder Fuß zu denken, jedoch ohne jedoch diese zu bewegen. Dieses Denken an eine Bewegung erfordert eine hohe Konzentrationsleistung um präzise Aktionspotentiale zur Computerprogramsteuerung zu generieren. Die Versuche werden hauptsächlich an unversehrten Personen durchgeführt. In Ausnahmefällen können auch Personen mit einer Querschnittslähmung hals-abwärts herangezogen werden (z.B. Cybathlon Piloten des Athena-Minerva Teams).

Die EEG-Biofeedback Experimente werden generell als sicher angesehen. Es gibt jedoch vereinzelt Berichte von unerwünschten Nebenwirkungen, wie z.B. Übelkeit, Konzentrationsstörungen und Schwindel. Falls solche Nebenwirkungen eintreten sollten, wird der Versuch abgebrochen.

4) Stellen Sie sicher, dass am Projekt beteiligte Personen keine Vorschädigungen mitbringen, die die Wirkung körperlicher und/oder psychischer Belastungen in gefährlicher Weise verstärken?

Eine ausreichende psychische Belastungsfähigkeit ist Voraussetzung zur Teilnahme an den Experimenten. Da man sich während den Experimenten möglichst wenig Bewegen soll, kommt es zu keinen physischen Belastungen, außer bei dem Auftragen des Gels mit stumpfen Spritzen auf die Kopfhaut.

5) Wenn Ihre Untersuchung bei Teilnehmer/innen Schmerz, psychischen Stress, Furcht, Erschöpfung oder andere negative Effekte hervorruft: Sind Sie sicher, dass ein über das im Alltag zu erwartende Maß nicht überschritten wird? Gibt es vergleichbare Studien?

Mit negativen Effekten über das alltägliche Maß hinaus ist nicht zu rechnen. Schmerzen, Stress und Erschöpfung werden nicht gezielt induziert. Erschöpfung kann jedoch wie bei jeder mentalen Konzentrationsleistung auftreten. Weiterhin können die stumpfen Nadeln auf der Kopfhaut einen leichten Schmerz verursachen, dass aber dann mit weniger Druck verhindert werden kann. Es gibt vereinzelte Berichte von unerwünschten Nebenwirkungen, wie z.B. Übelkeit, Konzentrationsstörungen und Schwindel. Die Daten werden mit einem 128 Elektroden EEG gemessen.

6) Haben Sie geprüft, ob es indirekte Beteiligte bzw. Betroffene des Vorhabens gibt (beispielsweise Angehörige durch Nachwirkungen von Stress oder Nebenwirkungen eingenommener Medikamente)?

Nach Prüfung ist niemand indirekt Betroffen.

7) Wenn personenbezogene Daten erhoben werden – sind es so wenig wie möglich, wurde hierzu informiert eingewilligt und wie ist der gesamte Lebenszyklus der Daten bis zu ihrer abschließenden Löschung kontrollierbar?

Es werden Gehirndaten mit einem EEG erhoben. Die Daten werden sicher und anonymisiert (mit fortlaufender Identifikationsnummer) gespeichert und lassen keine Rückschlüsse auf die Versuchsperson zu. Des Weiteren werden die Daten auf Festplatten gespeichert, welche ausschließlich dem Projektleiter und befugten Personen innerhalb des Cybathlon Teams der TU-Darmstadt zugänglich sind. Vor der Datenerhebung unterzeichnen die Probanden eine Einverständniserklärung auf der die Identifikationsnummer verzeichnet ist (nötig zur Datenlöschung). Diese Erklärung wird verschlossen am Institut aufbewahrt und nur der Projektleiter hat Zugang. Des Weiteren wird bestätigt, dass die Daten nur zu Forschungszwecken innerhalb des Teams genutzt werden und nicht an Dritte weitergegeben werden. Videodaten werden nicht erhoben.

8) Erfolgen die Einwilligungen schriftlich und wie werden die Einwilligungsdokumente sorgfältig aufbewahrt?

Ja.

9) Ist der Fall des Rücktritts vom Versuch – auch während der Durchführung – sowie die Löschung von Daten auf Wunsch von Versuchsbeteiligten vorgesehen?

Probanden können jederzeit das Experiment vorzeitig beenden. Eine Löschung der Daten auf Wunsch der Beteiligten ist jederzeit, insbesondere auch in späterer Folge, möglich.

10) Ist bei der Publikation der Forschungsergebnisse die Anonymität der Versuchsbeteiligten gewahrt? Werden auch nicht Minderheiten oder sozial verletzliche Gruppen – zum Beispiel durch die statistische Verknüpfung von Daten – in kollektiver Form bloßgestellt?

Anonymität wird im Rahmen der Veröffentlichung gewahrt. Probanden sind gesunde Personen und eventuell vom Hals abwärts Querschittsgelähmte. Nur der Projektleiter kann anhand der Identifikationsnummer und der unterzeichneten Einverständniserklärung die Identität der Versuchsperson feststellen. Teammitglieder und andere Autoren haben keinen Zugriff auf diese Informationen.

11) Geben Sie bitte in Stichworten an, welche Folgen die Forschungsergebnisse für die Gesellschaft haben können.

Schnelleres Erlernen der Beherrschung eines Brain Computer Interface Systems durch assistiertes Biofeedback. Insbesondere sollen BCI-Systeme alltagstauglich werden und innerhalb weniger Sekunden/Minuten einsatzbereit sein.

Sie können zu Ihrem Forschungsprojekt ein Votum der Ethikkommission der TU Darmstadt einholen. In diesem Fall bitten wir Sie, das **Antragsformular** auszufüllen, das auf der Webseite der Kommission [http://www.intern.tu-darmstadt.de/gremien/ethikkommisson/index.de.jsp] zu finden ist. Sie können aber auch die Ethikkommission – ohne ein Votum zu beantragen – lediglich über ihr Forschungsvorhaben informieren. In diesem Falle können Sie die diese **Checkliste** nutzen. Verwenden Sie dann die nachfolgenden Zeilen für eine Kurzbeschreibung Ihres Projekts und senden Sie die Liste der Ethikkommission zu:

| | |
|---|---|
| Verantwortliche Projektleitung/Ansprechpartner/in: | Dr. Elmar Rueckert |
| Emailadresse: | elmar@robot-learning.de |
| Titel des Vorhabens: | Adapting Brain Signals with Reinforcement Learning Strategies for BCIs |

Beschreibung der wesentlichen Züge der Projektdurchführung:

Dem Probanden wird eine EEG Kappe mit 128 Elektroden aufgesetzt. Zwischen Kopfhaut und Elektrode muss leitendes Gel gespritzt werden, damit möglichst wenig rauschen und möglichst aufschlussreiche neuronale Signale gemessen werden können. Nachdem die Kappe bereit ist, muss der Proband versuchen mit Hilfe seiner Gedanken eine digitale Spielfigur zu bewegen. Um dies zu schaffen, muss er an verschiedene Bewegungen mit Händen oder Füßen denken. Wenn dies erfolgreich ist, fängt die Figur an zu springen. In der ersten Phase hat der Proband 10 Minuten Zeit an verschiedene Bewegungen zu denken, um zu schauen, welche Bewegung die Figur am ehesten zum Springen bringt. In der Trainingsphase wendet der Proband diese Bewegung wiederholt in Gedanken an, um die Figur zum Springen zu bringen und Hindernissen auszuweichen. Anfangs wird der Proband beim Springen von einem Algorithmus unterstützt, der dem Probanden im Laufe der Zeit immer mehr eigene Kontrolle über die Spielfigur gibt. In der letzten Spielphase wird dann kein Biofeedback mehr benutzt, um die Figur zum Springen zu bringen, sondern der trainierte Algorithmus generiert rein durch den Probanden gesteuert Aktionspotentiale zur Spielsteuerung. In einer abschließenden Phase wird die Generalisierungsfähigkeit des trainierten Algorithmus anhand neuer Aufgaben (andere Hindernisse und schnelleres Spiel) getestet.

# Forschungsantrag

**Untersuchungstitel:** Adapting Brain Signals with Reinforcement Learning Strategies for Brain Computer Interfaces

**Untersuchungsleiter:** Dr. Elmar Rueckert

**Wissenschaftlicher Hintergrund**

Ein Problem heutiger Brain Computer Interface (BCI) Systeme ist, dass die Leistung des Probanden über die Zeit schnell abnehmen kann, da das Training lange dauert und sich die Signale über die Zeit verändern. Weiterhin kann die Motivation des Probanden schnell nach unten gehen, wenn er keinen Erfolg bei der Kontrolle eines BCI-Systems sieht.

**Ziel und Nutzen dieser Masterarbeit**

Ziel dieser Masterarbeit ist es zu untersuchen, welchen Einfluss assistiertes Neurofeedback durch Anwendung von Algorithmen aus dem Bereich maschinelles Lernen auf die Motivation des Probanden und die Lerngeschwindigkeit in der Kontrolle eines BCI Systems hat. Ein Erfolg dieser Thesis bei untersuchten Probanden könnte zu einer Verringerung der Trainingszeit führen, sowie eine Aussage darüber treffen, wie sehr hier angewandtes assistiertes Neurofeedback die Motivation des Probanden beeinflusst.

**Vorgehen**

In Zuge einer Studie am Institut für Intelligente Autonome Systeme, in Kollaboration mit dem Max-Planck Institut Tübingen, wird untersucht, ob es möglich ist, durch unterstützendes Feedback die Trainingsdauer in der Benutzung von Brain Computer Interface Systeme zu beschleunigen. Dabei wird einer Versuchsperson ein 128 Elektroden EEG auf den Kopf gesetzt. Um das Signal zu Rausch Verhältnis zu verringern, wird der Versuchsperson ein leitendes Gel zwischen Elektrode und Kopfhaut gespritzt.

Mit einem Brain Computer Interface System ist es möglich, Kommandos an einen Computer oder ein Exoskelett zu senden, um diesen/dieses zu steuern. Das Besondere hierbei ist, dass die Steuerung allein durch Gedankenkraft stattfindet.

In der ersten Phase des Versuches muss die Versuchsperson eine Spielfigur am Bildschirm dazu bringen zu springen. Hierbei bekommt die Versuchsperson ca. 10 Minuten Zeit, um in Gedanken verschiedene Bewegungen mit der bildlichen Vorstellung von Hand oder Fuß aus zu probieren. Die effektivste dieser Optionen wird in den weiteren zwei Phasen verwendet.

In der Trainingsphase, der zweiten Phase des Versuches, wird die Versuchsperson versuchen Steuerbefehle an den Computer zu senden um mit der Spielfigur Hindernissen auszuweichen. Die Versuchsperson wird anfangs von einer künstlichen Intelligenz unterstützt, die zu jedem Zeitpunkt die optimalen Befehl kennt. Mit zunehmender Trainingszeit bekommt der Proband bzw. die Probandin immer mehr eigene Kontrolle über die Figur und dominiert die Befehle der künstlichen Intelligenz, bis hin zur eigenständigen Kontrolle durch die Versuchsperson.

In der dritten und letzten Phase des Versuches wird das Trainingsergebnis, ein gelernter Klassifizierungsalgorithmus, anhand der Anpassungsfähigkeit an neue Spielabläufe getestet. Untersucht wird die Lernzeit um ein schnelleren Spielablauf zu meistern, bzw. um höheren Hindernissen ausweichen zu können.

**Mögliche Risiken**

Um eine gute Leitfähigkeit zwischen Elektrode und Haut zu erreichen, muss die Haut für das

Elektroenzephalogramm (EEG) aufgeraut und ein Leitungs-Gel appliziert werden. Dabei kann es eventuell zu Hautrötungen kommen, die nach einigen Stunden bis hin zu wenigen Tagen wieder abklingen. Das Elektrodengel lässt sich nach dem Versuch einfach mit Wasser und Seife entfernen.

EEG-Biofeedback, wie in dieser Studie durchgeführt, wird generell als sicher angesehen. Es gibt jedoch vereinzelte Berichte von unerwünschten Nebenwirkungen, wie z.B. Übelkeit, Konzentrationsstörungen und Schwindel. Sollten Sie während oder nach dieser Studie unerwünschte Nebenwirkungen an sich beobachten, so brechen Sie die Studie bitte unverzüglich ab und informieren Sie den Studienleiter.

**Datennutzung**
Im Rahmen dieser Studie werden EEG Daten von Ihnen erhoben. Diese Daten werden wie im folgenden beschrieben verwendet.

*Nutzung der Daten für Forschungszwecke im innerhalb des Cybathlon Teams der TU-Darmstadt*
Ihre Experimentdaten werden hauptsächlich von Mitgliedern des Cybathlon Teams der TU Darmstadt in pseudonymisierter Form, d.h. ohne Assoziierung mit Ihrem Namen, zu Forschungszwecken verwendet. Dies umfasst manuelle wie auch computergestützte Auswertung der oben genannten Daten.

*Nutzung der Daten für wissenschaftliche Publikationen und Vorträge*
Ihre Experimentdaten und die daraus gewonnen Erkenntnisse können in wissenschaftlichen Publikationen, auf Websites und bei wissenschaftlichen Tagungen aufgeführt oder veröffentlicht werden. Keinerlei persönliche Daten oder Daten, die Rückschlüsse auf die Versuchsperson zulassen, werden damit assoziiert.

**Vertraulichkeit der Daten**

Ihre Experiment-Daten werden sicher und pseudonymisiert (ohne Ihren Namen oder einen Verweis auf Ihre Identität) gespeichert. Ausschließlich dem Projektleiter und dem Projektkoordinator wird es möglich sein, die Daten auf die Identität der Versuchsperson zurück zu führen. Die Daten werden auf Festplatten gespeichert, welche ausschließlich befugten Mitarbeitern innerhalb der Forschergruppe der Einrichtung(en), deren Nutzung Sie zugestimmt haben, zugänglich sind.

Technische Universität Darmstadt | Karolinenplatz 5 | 64289 Darmstadt

TECHNISCHE UNIVERSITÄT DARMSTADT

# Information und Einverständniserklärung

## Projektleiter

Dr. Elmar Rueckert
Tel.: +49-6151-16-20074
Email: elmar@robot-learning.de

Sehr geehrte(r) Studienteinehmer(in), auf diesem Bogen finden sie zusätzliche Informationen und eine Einverständniserklärung über die Teilnahme an dieser Studie.

## Teilnahme

Ihre Teilnahme an dieser Studie ist völlig freiwillig. Sie können die Studie jederzeit beenden und die Löschung Ihrer Daten veranlassen.

## Risiken

Um eine gute Leitfähigkeit zwischen Elektrode und Haut zu erreichen, muss die Haut für das Elektroenzephalogramm (EEG) aufgeraut und ein Leitungs-Gel appliziert werden. Dabei kann es eventuell zu Hautrötungen kommen, die nach einigen Stunden bis hin zu wenigen Tagen wieder abklingen. Das Elektrodengel lässt sich nach dem Versuch einfach mit Wasser und Seife entfernen.

EEG-Biofeedback, wie in dieser Studie durchgeführt, wird generell als sicher angesehen. Es gibt jedoch vereinzelte Berichte von unerwünschten Nebenwirkungen, wie z.B. Übelkeit, Konzentrations-störungen und Schwindel. Sollten Sie während oder nach dieser Studie unerwünschte Neben-wirkungen an sich beobachten, so brechen Sie die Studie bitte unverzüglich ab und informieren Sie den Studienleiter.

## Datennutzung

Im Rahmen dieser Studie werden EEG Daten von Ihnen erhoben. Diese Daten werden wie im folgenden beschrieben verwendet.

### Nutzung der Daten für Forschungszwecke im innerhalb des Cybathlon Teams der TU-Darmstadt

Ihre Experimentdaten werden hauptsächlich von Mitgliedern des Cybathlon Teams der TU Darmstadt in pseudonymisierter Form, d.h. ohne Assoziierung mit Ihrem Namen, zu Forschungszwecken verwendet. Dies umfasst manuelle wie auch computergestützte Auswertung der oben genannten Daten.

### Nutzung der Daten für wissenschaftliche Publikationen und Vorträge

Ihre Experimentdaten und die daraus gewonnen Erkenntnisse können in wissenschaftlichen Publikationen, auf Websites und bei wissenschaftlichen Tagungen aufgeführt oder veröffentlicht werden. Keinerlei persönliche Daten oder Daten, die Rückschlüsse auf die Versuchsperson zulassen, werden damit assoziiert.

TECHNISCHE UNIVERSITÄT DARMSTADT

## Vertraulichkeit der Daten

Ihre Experiment-Daten werden sicher und pseudonymisiert (ohne Ihren Namen oder einen Verweis auf Ihre Identität) gespeichert. Ausschließlich dem Projektleiter und dem Projektkoordinator wird es möglich sein, die Daten auf die Identität der Versuchsperson zurück zu führen. Die Daten werden auf Festplatten gespeichert, welche ausschließlich befugten Mitarbeitern innerhalb der Forschergruppe der Einrichtung(en), deren Nutzung Sie zugestimmt haben, zugänglich sind.

## Einverstaendniserklaerung

Ich bestätige hiermit, dass ich die Informationen gründlich durchgelesen habe und über Wesen, Bedeutung, Tragweite sowie wissenschaftlichen Hintergrund der Studie informiert wurde. Außerdem wurde ich über eventuelle Risiken der Teilnahme an der Studie aufgeklärt. Ich hatte ausreichend Gelegenheit, mich über den Ablauf und Methodik zu informieren sowie im Zusammenhang mit der Studie auftretende Fragen zu stellen.

Ich wurde über den Umgang mit den erhobenen Daten sowie deren Speicherung informiert. Mir ist bekannt, dass ich mein Einverständnis jederzeit widerrufen und meine Daten löschen lassen kann.

Bei Fragen können Sie sich jederzeit an Dr. Elmar Rueckert wenden.
(Kontakt siehe oben)

**Ich bin damit Einverstanden, dass meine Experiment Daten zu Forschungszwecken innerhalb des Cybathlon Teams der TU-Darmstadt genutzt werden**

_____

Datum, Ort Unterschrift

**Ich bin damit Einverstanden, dass meine Experiment Daten für wissenschaftliche Veröffentlichungen und Präsentationen genutzt werden dürfen**

_____

Datum, Ort Unterschrift

Die Ethikkommission

TECHNISCHE UNIVERSITÄT DARMSTADT

## Antrag auf Begutachtung durch die Ethikkommission der Technischen Universität Darmstadt

| **Verantwortliche/r Wissenschaftler/in:** Elmar Rueckert | |
|---|---|
| Fachbereich: | Informatik, FB20 |
| Email: | elmar@robot-learning.de |
| Telefon: | +49-6151-16-20074 |
| Position: | Research Scientist |

**Angaben zum Forschungsvorhaben**

| | |
|---|---|
| (Arbeits-)Titel: | Adapting Brain Signals with Reinforcement Learning Strategies for BCIs |
| Kurze Beschreibung: | Verbesserung der Trainingsdauer zur Nutzung von Brain Computer Interface Systemen durch unterstuetzendes Feedback von kuenstlichen Agenten. |
| Geplante Dauer: | 6 Monate |
| Beteiligte Wissenschaftler/innen: | David Sharma, Prof. Dr. Moritz Grosse-Wentrup, Prof. Dr. Jan Peters, Dr. Elmar Rueckert |
| Beantragte Drittmittel, Höhe und Geldgeber: | - |
| Votum der Ethikkommission für Antragstellung notwendig: | ja ☒<br>nein ☐ |

Bitte beschreiben Sie kurz, welche gesetzlichen Vorschriften und Standesregeln einschlägig sind. Erläutern Sie bitte auch, wie Sie diese einhalten.

Wir halten uns an das hessische Datenschutzgesetz.

Bitte formulieren Sie Ihr Forschungsvorhaben, insbesondere den Versuchsaufbau, allgemein verständlich:

In Zuge einer Studie am Institut für Intelligente Autonome Systeme, in Kollaboration mit dem Max-Planck Institut Tübingen, wird untersucht, ob es möglich ist, durch unterstützendes Feedback die Trainingsdauer in der Benutzung von Brain Computer Interface Systeme zu beschleunigen. Dabei wird einer Versuchsperson ein 128 Elektroden EEG auf den Kopf gesetzt. Um das Signal zu Rausch Verhältnis zu verringern, wird der Versuchsperson ein leitendes Gel zwischen Elektrode und Kopfhaut gespritzt.

Mit einem Brain Computer Interface System ist es möglich, Kommandos an einen Computer oder ein Exoskelett zu senden, um diesen/dieses zu steuern. Das Besondere hierbei ist, dass die Steuerung allein durch Gedankenkraft stattfindet.

In der ersten Phase des Versuches muss die Versuchsperson eine Spielfigur am Bildschirm dazu bringen zu springen. Hierbei bekommt die Versuchsperson ca. 10 Minuten Zeit, um in Gedanken verschiedene Bewegungen

mit der bildlichen Vorstellung von Hand oder Fuß aus zu probieren. Die effektivste dieser Optionen wird in den weiteren zwei Phasen verwendet.

In der Trainingsphase, der zweiten Phase des Versuches, wird die Versuchsperson versuchen Steuerbefehle an den Computer zu senden um mit der Spielfigur Hindernissen auszuweichen. Die Versuchsperson wird anfangs von einer künstlichen Intelligenz unterstützt, die zu jedem Zeitpunkt die optimalen Befehl kennt. Mit zunehmender Trainingszeit bekommt der Proband bzw. die Probandin immer mehr eigene Kontrolle über die Figur und dominiert die Befehle der künstlichen Intelligenz, bis hin zur eigenständigen Kontrolle durch die Versuchsperson.

In der dritten und letzten Phase des Versuches wird das Trainingsergebnis, ein gelernter Klassifizierungsalgorithmus, anhand der Anpassungsfähigkeit an neue Spielabläufe getestet. Untersucht wird die Lernzeit um ein schnelleren Spielablauf zu meistern, bzw. um höheren Hindernissen ausweichen zu können.

Beigefügt sind:

☒ Ausgefüllte Checkliste

☒ Forschungsantrag

☒ Einverständniserklärungsformular der Teilnehmenden

☐ ggf. Fragebogen

☐ sonstiges: Klicken Sie hier, um Text einzugeben.

Datum und Unterschrift der/des verantwortlichen Wissenschaftler/in (AG-Leiter/in, FG-Leiter/in)[1]

[1] DFG-antragsberechtigt

## A.3 Experiment Instruction

**Anleitung**

**Einleitung**
Vielen Dank, dass Sie an diesem Experiment teilnehmen. Im folgenden Dokument beschreibe ich, was auf Sie zukommt. Im Allgemeinen geht es darum, zu erlernen eine Spielfigur mit Gedanken steuern zu können. Dazu benutzen wir ein Elektroenzephalografie-System (EEG) und entsprechende Software, die mit dem Spiel kommunizieren kann, um Befehle an das Spiel zu senden. Dieses System nennt sich Brain-Computer Interface.

**Vorbereitung**
Damit wir Hirnsignale von Ihnen Messen können, müssen wir Ihnen eine EEG-Kappe mit 128 Elektroden aufsetzen. Zwischen Kopfhaut und jeder Elektrode muss ein leitendes Gel gespritzt werden, um das Signal zu Rausch Verhältnis zu verringern. Die Vorbereitung kann 30-60 Minuten dauern.

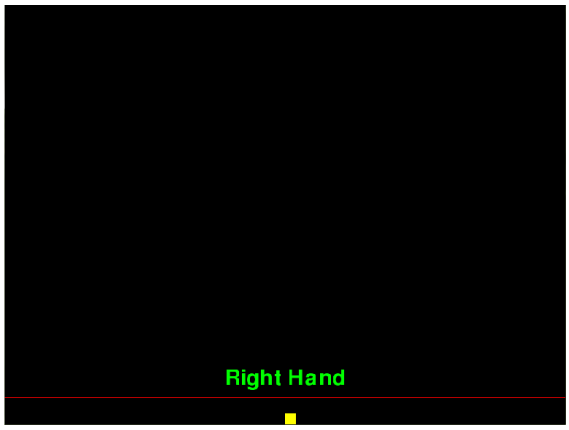
**Motor Imagery**
Motor Imagery bedeutet, dass alleine durch das Denken einer Bewegung die gleichen Hirnareale in ähnlicher Weise aktiviert werden, wie wenn die Bewegung wirklich ausgeführt wird. Zusätzlich zum Gedanken an die Bewegung sollte man auch versuchen, die Bewegung zu „spüren“, also sich vorzustellen, wie sich die Bewegung anfühlt, z.B. wenn man einen Stoffball in der Hand wiederholt zusammen drückt. Wenn dieses Denken und Fühlen der Bewegung funktioniert, dann werden entsprechende Hirnareale aktiviert und man könnte z.B. eine Spielfigur damit steuern, wie in diesem Experiment geplant.

**Das Spiel**
Das Spiel ist sehr einfach gehalten. Es besteht nur aus einer Spielfigur und einem Hindernis, welches auf die Spielfigur zukommt. Ziel ist es, mit der Spielfigur das Hindernis mithilfe von Motor Imagery zu überspringen. Jedes Mal, wenn die Spielfigur das Hindernis überspringt, gibt es einen Bonus von 400 Punkten, wenn sie mit dem Hindernis kollidiert, dann eine Bestrafung von -50 Punkten. Zusätzlich gibt es für jeden Sprung einen Minuspunkt. Damit soll verhindert werden, dass der Proband versucht die Figur von Anfang bis zum Ende oben zu halten. Vielmehr soll versucht werden, dass richtige Timing für den Sprung gefunden zu werden, so dass man mit möglichst wenig Sprüngen über das Hindernis kommt.

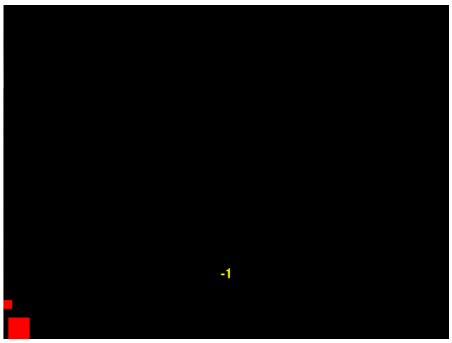
**Verschiedene Spielphasen**
Es gibt insgesamt drei Phasen während Sie Spielen:

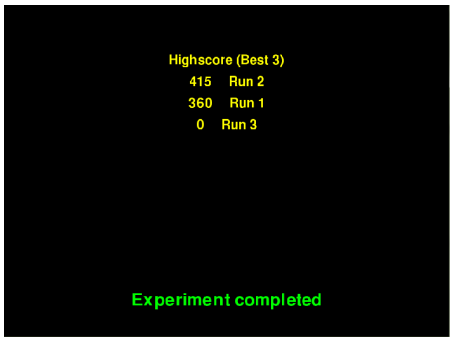
- **Entspannungsphase:** In dieser Phase sollten Sie entspannen. Das heißt, an keine Bewegungen denken. Außerdem ist es wichtig, dass Sie ruhig sitzen bleiben und sich nicht real bewegen, da dies sonst zu unerwünschten Artefakten in den Signalen führen könnte. Diese Phase dauert ca. 6 Sekunden.
- **Aktivitätsphase:** In der Aktivitätsphase müssen Sie an eine Bewegung mit der rechten Hand, oder mit der linken Hand oder mit den Füßen denken. Wichtig ist, dass Sie diese Bewegung in Gedanken ständig wiederholen. Auch hier sollte man ruhig sitzen bleiben und sich nicht real bewegen. Diese Phase dauert ca. 6 Sekunden.
- **Pause Phase:** In dieser Phase können Sie sich bewegen, wenn Sie wollen. Auch diese Phase dauert ca. 6 Sekunden.

**Erste Phase des Experiments (Ausprobieren)**
In der ersten Phase des Experiments gibt es nur die Spielfigur und eine rote Linie. Ziel ist es, die Figur mit Motor Imagery über die Linie zu kriegen, die daraufhin grün wird. Der Sinn dieser Phase ist es, Ihnen die Möglichkeit zu geben, verschiedene Bewegungen in Gedanken mit den jeweiligen Extremitäten auszuprobieren und dann pro Extremität eine Bewegung zu wählen, die dann in der nächsten Phase des Experiments benutzt werden kann. In dieser Phase des Experiments sind alle vorher beschriebenen Spielphasen vertreten. Diese Phase dauert ca. 10 Minuten.




**Zweite Phase des Experiments (Anwenden)**
Die zweite Phase des Experiments ist die Trainingsphase. Hier werden die zwei besten Extremitäten aus vorheriger Phase benutzt. Im Spiel kommt ein Hindernis auf Sie zu, welches Sie mit vorher ausprobierten Bewegungen ausweichen sollen. Auch hier sind alle drei Spielphasen vertreten. Ziel dieser Phase ist es zu versuchen, Ihre Hirnsignale so anzupassen, dass Sie das Gefühl kriegen, die Figur mit Ihren Gedanken kontrollieren zu können. Wie schon vorher erwähnt, wird jeder Sprung mit einem Minuspunkt bestraft. Deswegen ist es wichtig, das richtige Timing für den Sprung zu finden, sodass Ihnen nicht zu viele Punkte abgezogen werden. D.h. bleiben Sie am Anfang der Aktivitätsphase etwas länger im entspannten Zustand, damit die Figur möglichst unten bleibt. Bitte beachten Sie, dass es etwas dauern kann, bis Ihre Signale vom Algorithmus erkannt werden. Es wird insgesamt 8 Runs mit jeweils 20 Hindernissen geben. Nach jedem Run wird Ihnen eine Highscore der besten 3 Ergebnisse der bisherigen Runs angezeigt. Sie werden mit der Punktzahl in der Highscore nie unter 0 kommen können. Diese Phase dauert ca. 1 Stunde. Während den ersten 4 Runs gibt es nach jedem Run eine Pause von einer Minute. Danach gibt es bis zum Ende nach jedem Run eine Pause von zwei Minuten.




**Dritte Phase des Experiments (Generalisierungsfähigkeit)**
Die dritte und letzte Phase des Spiels ist ähnlich zur zweiten Phase des Spiels. Der Unterschied ist, dass Hindernisse auf Sie zukommen, die Sie vorher noch nicht gesehen haben, z.B. höhere, breitere oder schnellere Hindernisse. Hier wird untersucht, ob Sie das Brain Computer Interface System nach dem Training schon so gut kontrollieren können, dass Sie Ihre Hirnsignale an das veränderte Hindernis anpassen. In dieser Phase gibt es die Entspannungsphase und die Pause Phase nicht mehr. Diese Phase dauert ca. 20 Minuten.

**Hinweise:**
Auch wenn es schwer ist, versuchen Sie Ihren Emotionen unabhängig vom momentanen Verlauf des Versuches zu kontrollieren, da Emotionen Ihre Leistung sowie Ihre Hirnsignale stark in negativer Weise beeinflussen können und Sie eher ablenkt. Konzentration ist sehr wichtig bei diesem Versuch. Falls Freude oder Ärger aufkommt, versuchen Sie sich schnell wieder zu beruhigen.

**Ziel des Experiments**
Ziel des Experiments ist es zu untersuchen, ob es mit meiner Methode möglich ist ein Brain Computer Interface System schneller zu kontrollieren, als mit herkömmlichen Trainingsmethoden.

Falls Sie Fragen haben, wenden Sie sich bitte an den Versuchsleiter.