Radboud University

BSc thesis

Modelling BCI Learning

Author:
Daan Roos
d.roos@student.ru.nl
+316 361 691 44
s4373596

Supervisor:
J.D.R Farquhar
M.R. van der Waal

August 23, 2017

Contents

Abstract

The aim of this thesis is to investigate if it is possible to computationally model BCI learning and use reinforcement learning to predict the effects of different neurofeedback parameters. A model of BCI learning was constructed and experiments were run with Q-learning agents using different model parameters. According to the model, continuous feedback leads to significantly more efficient BCI learning than discrete feedback, and BCI learning becomes quadratically slower as the signal-to-noise ratio decreases. Optimistic feedback results in significantly faster learning than neutral feedback, and pessimistic feedback significantly slows down learning relative to neutral feedback. Since research in this area is rather scarce, it is hard to compare the results gained using the model with empirical data. However, the results do seem to reflect existing data. The optimistic feedback data reflecting empirical data seems to be the result of a side-effect of the way rewards are calculated. Overall, the results suggest that the model may be used to predict the effects of different neurofeedback parameters, but further research would be needed.

1 Introduction

Brain-Computer Interfacing has a lot of potential. Among others, it can help people with locked-in syndrome to communicate [1], it can give disabled patients the ability to control a neuroprosthetic [2], and it has a variety of applications for healthy users [3]. Unfortunately, controlling a BCI still requires a lot of training [4]. Research on how to best train users is still rather scarce [5], so with this thesis I try to answer some questions on how to best give feedback during BCI learning.

In this thesis, I investigate if it is possible to construct a computational model of BCI learning and use reinforcement learning to predict the effects of different neurofeedback parameters. First, background information on Brain-Computer Interfacing, neurofeedback, and reinforcement learning will be given. The construction of the model will be explained and how it was used to investigate the effects of different neurofeedback parameters. Then, the results of the experiments are given and these results are discussed. This thesis ends with a conclusion of the knowledge gained with this research.

1.1 Brain-Computer Interfacing

Advances in neuroscience, signal processing, and machine learning have made it possible to derive a user's intention using only their brain state [2]. We call a device that accomplishes this a Brain-Computer Interface (BCI). A Brain-Computer Interface allows a user's brain state to be translated into commands that can control an external device [1]. While initial research to the use of BCIs was mainly focused on communication and control for patients who are completely paralysed [2], it has been shown that BCIs can be applied in a wide variety of ways, including, but not limited to, helping patients with ADHD or epilepsy [6], post-stroke rehabilitation [7], and even controlling video games [8]. One of the long-term goals of BCI research is to allow disabled users to control a robotic arm or neuroprosthetic. At first it was believed that this would only be possible using invasive BCIs, which derive the user's intent from potentials recorded within the brain [9]. In 2004 however, Wolpaw and McFarland showed that multidimensional control is also possible using noninvasive BCIs [2]. This made BCI research much more relevant to a lot of potential users, because invasive BCIs have a lot of disadvantages [10].

Since BCIs make use of a person's brain state, an important BCI component is its measurement technology. Different measurement technologies are used to measure brain activity, such as electroencephalography (EEG) [2], magnetoencephalography (MEG) [11], functional magnetic resonance imaging (fMRI) [12], and functional near-infrared spectroscopy (fNIRS) [13]. Most current BCIs use EEG because of its high temporal resolution, ease-of-use, and relative low costs [9]. This research focuses on EEG, for more information on other measurement technologies see [9]. EEG measures electric brain activity on the scalp, caused by the flow of electric currents during synaptic excitations of the dendrites in neurons [14]. A BCI tries to extract characteristics out of these EEG signals that are uniquely tied to a specific mental process or state [15]. These characteristics are called *signatures*. Specific mental tasks are used to generate such signatures [9].

Our focus is on Event-Related Desynchronization and Synchronization (ERD/ERS) signatures. ERD and ERS correspond to amplitude modulations in the power of EEG signals in certain frequency bands [16]. An ERD results in a smaller amplitude in EEG signals, whereas an ERS results in a larger amplitude. ERD and ERS can be elicited using specific mental tasks, and can be observed in different spatial locations depending on what mental task is used [17]. This makes them an excellent signature for BCI control. One of the most-used mental tasks to elicit ERD/ERS is imagined movement (IM) [18]. Movement and imagined movement result in an ERD and ERS in the mu frequency band (8-12Hz) and the beta frequency band (16-24Hz) [19]. Both ERD and ERS for movement imagery can be found in the contralateral sensorimotor cortex (i.e. on the opposite side of the imagined limb movement). Thus, based on the location of the ERD/ERS a BCI can discriminate between different mental tasks, which can then be translated into actions [19]. Other mental tasks that elicit ERD/ERS include mental rotation, word association, auditory imagery of a melody, and mental subtraction [20]. Not all mental tasks result in equal BCI performance for all users [21]. Mental tasks related to a person's specific skill might enhance performance, so the optimal tasks differ for each user [22]. Currently, BCIs that make use of ERD/ERS can be used with up to four different mental tasks. More than four tasks, and the prediction accuracy and thus BCI performance decreases [23]. This means that it is currently possible for a user to control a wheelchair using these four tasks, by linking each task to a specific command (e.g. right hand IM → forward).

Brain-Computer Interfacing has amazing potential. It gives patients who are unable to move any muscles a method of communication [1], it can allow disabled patients to control a robotic arm or neuroprosthetic [2], and it can even be useful for healthy users by monitoring mental workload [24] or allowing control of video games [8]. Currently, it still takes a lot of training for users to become proficient at controlling a BCI [23]. This means that the way we train users becomes very important. Unfortunately, not a lot of research has been done in this area [4]. Currently, using a BCI still requires expensive equipment, trained professionals to handle the equipment, and a lot of training time. If it turns out to be possible to computationally model BCI learning and use this model to investigate the efficacy of different neurofeedback parameters, this could save researchers a lot of time and money, and potentially benefit a large amount of people.

1.2 Neurofeedback

In the 1960s it was shown that it is possible for humans to gain control over their own brain waves through the use of instantaneous feedback [25]. Using instantaneous feedback and operant conditioning, users are able to learn how to control different electrophysiological components of their brain activity [26]. The goal of neurofeedback is to teach users what specific states of cortical arousal feel like, and how to activate such states voluntarily [27]. Neurofeedback has been shown to be effective at improving symptoms of children with autism [28], ADHD and epilepsy symptoms [6], and schizophrenia [29], among others. Neurofeedback is a form of BCI, since brain activity is measured, processed, and output is fed back to the user.

In a typical neurofeedback setting, a bar or cursor is used to give feedback on the user's brain state [27]. If the user's brain state is desirable, the bar will grow or the cursor will move towards a certain stimulus. For example, a user will be shown a bar whose size represents the amplitude of a certain frequency. The user will then be given the goal to try and maximise the length of the bar. If the bar reaches its maximum height, a tone may sound or a counter will increase. This tone and counter increase act as a reward for the user. This same protocol is repeated until the user has learnt the BCI skill. The user should be able to consciously modulate the size of the bar, and thus the amplitude of that certain frequency in their own brain.

Wolpaw et al. (2002) have shown that BCI use is a skill, and as such it can be learned [1]. Neuper and Pfurtscheller (2010) have proven that neurofeedback is a necessary component to learn the BCI skill [30]. Lotte et al. (2013) argue that the user is one of the most critical components BCI loop [5]. Improving the training paradigms (i.e. neurofeedback) used in BCI skill acquisition may be one of the easiest, yet powerful methods to improve BCI performance, and yet there has hardly been any research in this direction. Even the so-called "BCI illiterates", about 20% of users who are unable to learn how to control a BCI, may be explained by flaws in current BCI training approaches [5]. This gives another reason why investigating the optimal way to give feedback in BCI skill acquisition is a good idea.

1.3 Reinforcement Learning

Reinforcement Learning (RL) is a form of machine learning. Reinforcement learning is learning how to behave in certain situations to maximise a numerical reward [31]. The learner is not told how to behave, as in supervised learning, but instead must discover what actions lead to the highest rewards by trying them. This means that the learner must perform a trial-and-error search of the possible actions to discover which actions are the most effective. This becomes harder when actions may initially seem very beneficial due to a high immediate reward, while other actions may be more beneficial in the long-term. Reinforcement learning can be characterised as "learning by doing", since the learner needs to learn how to behave by performing random actions and evaluating the outcomes of these actions.

We call a reinforcement learner a learning *agent*. An agent interacts with its environment by performing actions. These actions affect the agent's own state and the environment. An agent must also have a goal it can work towards, such as maximising a numerical reward. Rewards are returned by the environment when the agent performs an action. Reinforcement learning is often used in environments that are interactive, since other methods such as supervised learning are often inadequate for learning from interaction [31]. Reinforcement learning agents should be able to learn how to interact with uncharted territory, since in such areas learning from the environment is most important.

One of the dilemmas in reinforcement learning is the trade-off between *exploration* and *exploitation* [31]. When an agent is exploring its environment, it performs random actions to get a sense of the rewards returned by those actions. However, if an agent is constantly exploring, it has no chance to benefit from the newly gained knowledge. When an agent is exploiting, it is selecting the actions that it knows are good, it is performing *greedy* actions. But when an agent is only exploiting known actions, it may miss actions that are more beneficial. The agent must try a variety of actions, and favour the ones that seem best.

Most reinforcement learning environments are formulated as a Markov Decision Process (MDP) [32]. An MDP is a mathematical framework for modelling decision making in situations where outcomes are are partly random and partly under the control of a decision maker [33]. An MDP consists of a set of states S, a set of actions A, a set of probabilities $P_a(s, s')$ of state s leading to state s' after action a, a set of rewards $R_a(s, s')$ with the rewards of action a used in state s leading to state s', and a discount factor γ, which represents the importance of future rewards relative to present rewards. The main problem of an MDP is finding a policy π, that specifies for each state what action to perform to maximise the total reward. This problem can be solved using reinforcement learning.

In this research, temporal-difference (TD) learning is used. This is a category of reinforcement learning algorithms that do not keep an internal model of the environment

[31]. TD methods use a state value function, which contains a value for each state encountered. The value of each state is updated using the prediction error, the difference between the expected reward and the actual reward. Repeated updating of these values leads to increasingly better predictions, and thus to better performance. Rushworth and Behrens (2008) have shown that this update rule can be used to predict choices in animals [34], so investigating whether we can use this same update rule to predict BCI learning is an interesting next step.

1.4 Research Questions

In our research group we research how to best give feedback as to optimise the BCI learning rate. Specific questions regarding BCI learning that are investigated are: the effect of continuous vs. discrete feedback by Jordy Ripperda, the effect of optimistic and pessimistic feedback by Max Moons, and the effect of motivation by Sjoerd Bos. They will research these questions with real-life experiments using EEG. My research will focus on the question whether we can use reinforcement learning to predict these same effects. I will create a computational model of how a person learns to use a BCI, and then use a reinforcement learning algorithm to compare the learning rates of the agent using different model parameters. The agent's results will then be compared with the results of the real-life experiments by the other members of the group, and existing literature. Since EEG is very susceptible to noise [35], the effect of EEG noise on BCI learning will also be investigated, as well as the effect of noise on the two different types of feedback.

My research question will therefore be: **Can we use a reinforcement learning model to predict the effects of different neurofeedback parameters on BCI skill acquisition?**

My sub-questions are related to the research questions of my group members. I will compare the model's results with empirical data, and compare the model's results using different parameter settings under the assumption that the model reflects empirical results.

- Do the results of the model reflect empirical results?
  - Does the effect of continuous vs. discrete feedback reflect empirical results?
  - Does the effect of optimistic vs. neutral vs. pessimistic feedback reflect empirical results?
- Assuming the model reflects empirical results, what is the effect of different neurofeedback parameters on BCI skill acquisition?
  - What is the effect of continuous vs. discrete feedback on BCI skill acquisition?
  - What is the effect of optimistic vs. neutral vs. pessimistic (continuous) feedback on BCI skill acquisition?

  - What is the effect of noise on BCI skill acquisition?

1.5 Hypotheses

1.5.1 Continuous vs. discrete feedback

Neuper et al. (1998) suggest that continuous feedback leads to more efficient learning [36]. McFarland et al. (2008) however, showed that continuous feedback can be facilitory or inhibitory to BCI skill acquisition, depending on the user [37]. Continuous cursor movement is a form of direct feedback, which if the user is doing well, can work reinforcing. If the subject is not performing well, this continuous movement might be distracting, leading to worse motivation and performance. Additionally, the continuous cursor movement might cause certain EEG responses that interfere with BCI control, or might lead to eye movement artifacts that distort the EEG signal [37]. So there are a lot of different effects related to continuous and discrete feedback that might influence BCI skill acquisition in humans. Effects such as motivation and EEG responses will not be investigated, since these are too complex to model. Because reinforcement learning agents learn from the rewards gained by actions, it can be expected that continuous feedback will be more useful to the agent, since this gives direct feedback on the value of actions. Whereas discrete feedback at the end of an episode will be feedback on the entire trial, and the agent will have to find out which actions over the entire trial lead to a good outcome. The agent will be able to do this due to the discount factor [31], but it will take a large amount of episodes depending on the amount of states in the episodes (the more states in an episode, the harder it will be for the agent to deduce which actions are positive). It is hypothesized that continuous feedback in the model will lead to faster learning, whereas in the human trials this will depend on the user.

1.5.2 Optimistic vs. neutral vs. pessimistic feedback

Motivation has been shown to play a large role in BCI skill acquisition [22]. Barbero and Grosse-Wentrup (2010) suggest that incapable subjects benefit from positively biased feedback, whereas already capable subjects perform worse given inaccurate feedback [38]. Negative feedback has been shown to decrease motivation [39]. Barbero and Grosse-Wentrup hypothesize that poor performance (which can seem random to the user) demotivates the user which leads to even poorer performance. In this case, biased feedback will keep the user's motivation up until they become more proficient at controlling the BCI. Capable users are already motivated since their performance is high, so they benefit the most from accurate feedback to improve their skill. As already mentioned, motivation will not be modeled, so this factor will not have an influence on learning performance. Therefore, in the model, optimistic and pessimistic feedback will essentially just be noisy rewards in one direction (i.e. positive or negative). This suggests that optimistic and pessimistic feedback will lead to slower learning, and both will decrease learning performance equally.

1.5.3 Noise and BCI learning

Noise in an EEG system distorts the signal and thus makes it harder to extract the brain signals from the EEG signal [40], leading to bad classification performance. So in real-life experiments, noise should be minimised. In the model, noise essentially leads to noisy rewards. The agent can perform a correct action, but due to noise the wrong mental task might be predicted and thus a bad reward will be returned. Therefore, it is to be expected that the more noise is present in the system, the worse learning performance will be. Additionally, noise might have a different effect on continuous and discrete feedback. Since the noise in this research will be gaussian, it may be expected that it has a smaller effect on discrete feedback due to the noise canceling out at the end of a trial. Whereas continuous feedback will lead to constant noisy rewards. So it is hypothesized that the effect of noise is bigger in the continuous feedback condition than in the discrete feedback condition.

2 Methods

2.1 Modeling BCI Learning

To investigate if it's possible to model BCI learning, and to see if that model can predict the empirical effects of different neurofeedback parameters, first a model had to be built. A model can be defined as a simplified version of reality. This means that even though some parts of the real-life process will be abstracted, the goal is to create a meaningful model that can be used to predict empirical data. In this case, the goal is to model the way users learn to control a BCI using the Buffer BCI system.

During a Buffer BCI training session, the user will see a black screen with a number of ellipses. One ellipse is shown for each task (e.g. imaginary feet movement), each on a circle with equal amounts of space in-between them. In the center of each ellipse a label specifies what task is represented by that ellipse. In the center of the screen, a cursor ellipse is shown. These ellipses are used to give feedback to the user. The Buffer BCI system training sessions consists of trials, and each trial has a task. At the beginning of a trial, the task for that trial is shown to the user at the center of the screen. During the trial the user will perform said task (e.g. think about moving their feet), and the system will give feedback based on its prediction. This feedback is either continuous or discrete. In the case of continuous feedback, the cursor constantly shows what the current prediction is using its location (i.e. the cursor being closest to the ellipse labeled 'Feet' means the system predicts the user is imagining moving their feet). At the end of a continuous feedback trial, the ellipse that was closest to the cursor will change color. The task represented by that ellipse is the task that has been predicted by the system. In the case of discrete feedback, the cursor is not shown the entire trial. Only at the end of the trial feedback is given, by coloring the ellipse with the task that was predicted based on the input of the entire trial. See figure 1 for a visualisation of a Buffer BCI discrete feedback trial. After enough training sessions, a user should have learned how

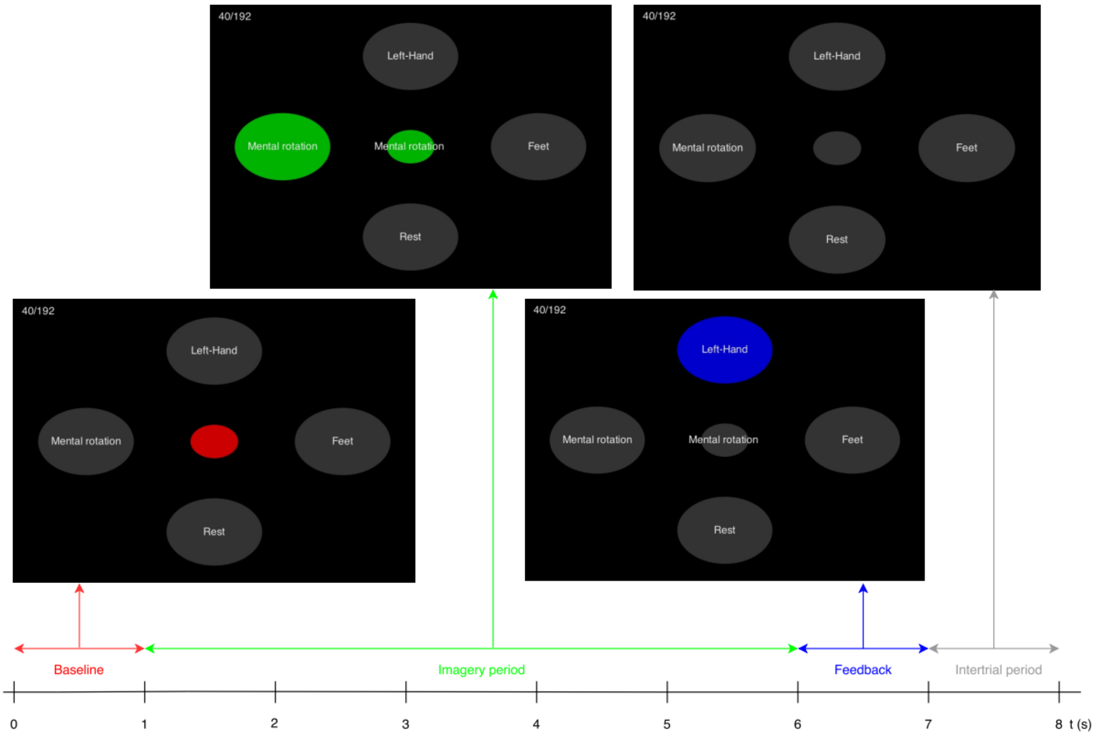

Figure 1: Visualisation of a Buffer BCI discrete feedback trial. The baseline phase presents the user with the trial task. During the imagery period the user performs this task. In the feedback phase the system will present the user with the predicted class. After this, there is an intertrial period before the next trial starts. Image by Jordy Ripperda.

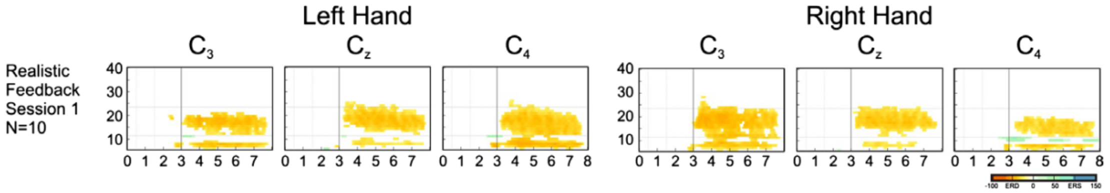

Figure 2: Time-frequency representation of ERD/ERS over the C3, Cz and C4 electrodes. The x-axis represents time in seconds (s). The y-axis the frequency range 4-40Hz. Adapted from Neuper et al. (2009) [41].

to control the cursor in the case of continuous feedback, and how to produce correct classifications in the case of discrete feedback.

Certain requirements for this model followed from the research questions: 1) the model should have some measure of brain activity, 2) the model has to be able to represent different tasks, 3) the model should be able to represent noise, 4) the model should be a Markov Decision Process. To satisfy these requirements, the following model was constructed.

2.1.1 States

In the type of BCI that we study, different tasks have different spatio-temporal characteristics. Imagined right hand movement leads to an ERD over the contralateral (i.e. left) sensorimotor cortex, while imagined left hand movement leads to an ERD in the right sensorimotor cortex [42]. These ERDs can last up to several seconds (see Figure 2). Based on the spatio-temporal characteristics of these different tasks, the BCI can discriminate the task that was being performed by the user. The more a user trains using a BCI, the better the user will be able to control these ERDs [2].

Since it is not possible to model the entire brain, it was necessary to find an abstraction for the different spatio-temporal characteristics of the different tasks. This was done by representing the activity of each task with a continuous value between 0.0 and 1.0, i.e. 0% to 100% activity on a certain task. Just like a user can have a strong ERD over the left primary sensorimotor cortex and no ERD over the right primary sensorimotor cortex, an agent can have 100% activity on one task (e.g. right hand IM), but 0% activity on a different task (e.g. left hand IM) .

The agent learning the model needs to learn what action to perform for each state it encounters. And what action needs to performed is dependent on the task of the current trial. So the states also store the task of the current trial (e.g. right hand IM). Since one of the research questions involves the effect of noise on BCI skill acquisition, there should be some measure of noise in the states. Noise in an EEG system lowers the signal-to-noise ratio, which makes it harder to correctly predict the task the user is

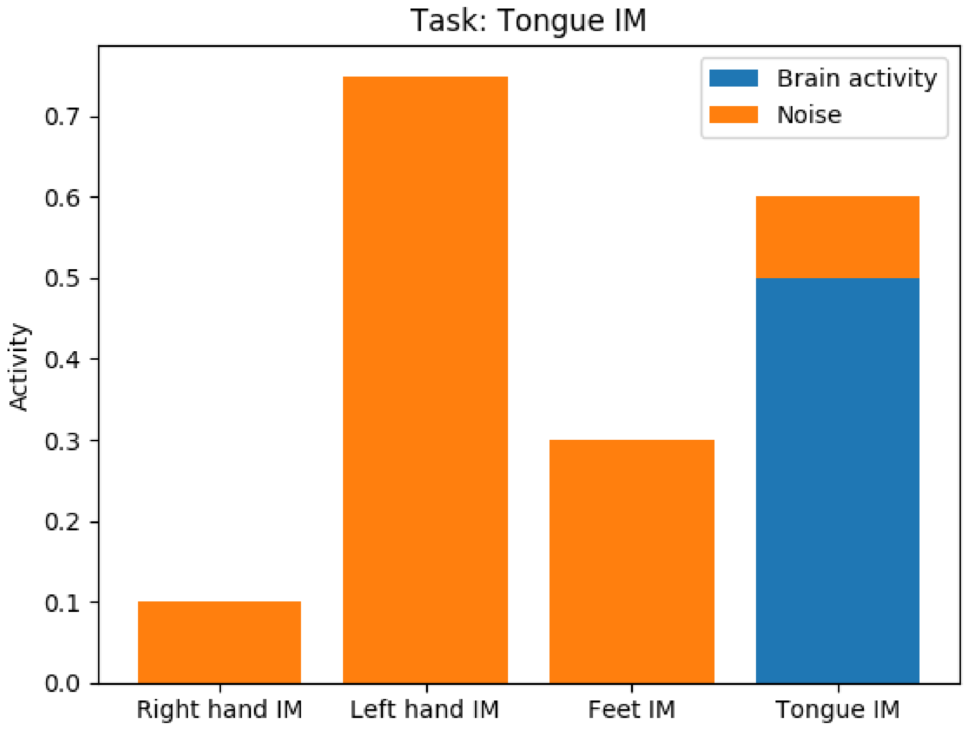

Figure 3: Visualisation of a state in the model. A state consists of a task (e.g. tongue IM), a continuous value between 0 and 1 for each task, representing the activity for that task, and a continuous noise value for each task. In this example state, because of the high amount of noise the system will predict the agent is performing left hand imagined movement. The agent however, has higher activity in the imagined tongue movement task, which is also the trial task it is supposed to be doing. Even though the agent is performing the correct task, it will have trouble learning because the system will predict it is performing left hand imagined movement.

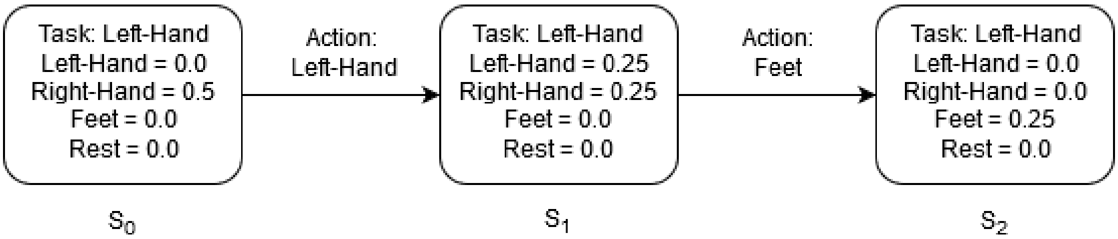

Figure 4: Visualisation of the effects of different actions on states. When a certain task is performed as action, the activity of that task will go up with the modulation rate, while the activity of the other tasks gets lowered with the modulation rate. In this example, the modulation rate is set at 0.25.

performing [40]. In the model this is emulated by having a continuous value for each task on top of the task activity. This means that as the noise increases, the chance of misclassification increases, the rewards will get noisier, and thus learning will be slower. This is effectively the same as the signal-to-noise ratio in EEG. The agent only has control over the task activity (by performing actions), and noise gets added by the environment. The rewards are based on the task activity and the noise. For a visualisation of the states, see Figure 3.

2.1.2 Actions

In a real-life BCI session, the user will perform a certain task to induce an ERD, which is then used by the BCI system to predict which task is being performed. The model works in the same way. The agent can select a task from all possible tasks, and the brain activity value of that task will increase with the value of a parameter named the *modulation rate*. Even though the brain is incredibly complex and has a staggering amount of processing capacity, humans are still limited to being focused on one thing at a time [43]. For this reason, when the agent performs a task, the activity of that task will increase with the modulation rate, while the activity of the other tasks will decrease with the modulation rate. See figure 4 for a visualisation of how different actions affect activity of a state. The actions in this model are deterministic or stochastic dependent upon the noise. If there is no system noise, the actions are deterministic: the same action in the same state will always have the same effect. However, when noise *is* present, the environment is stochastic, since there's an element of randomness to the noise. Even though the noise is random, the model does satisfy the Markov Property, the conditional probability distribution of the next state is only dependent on the current state and the parameter settings of the model. These parameters do not change during experiments.

The amount of noise in the states is determined by the *system noise* parameter. For every action that is performed, a random gaussian value gets drawn for each task, and is multiplied with the system noise parameter. This value is then added to the current

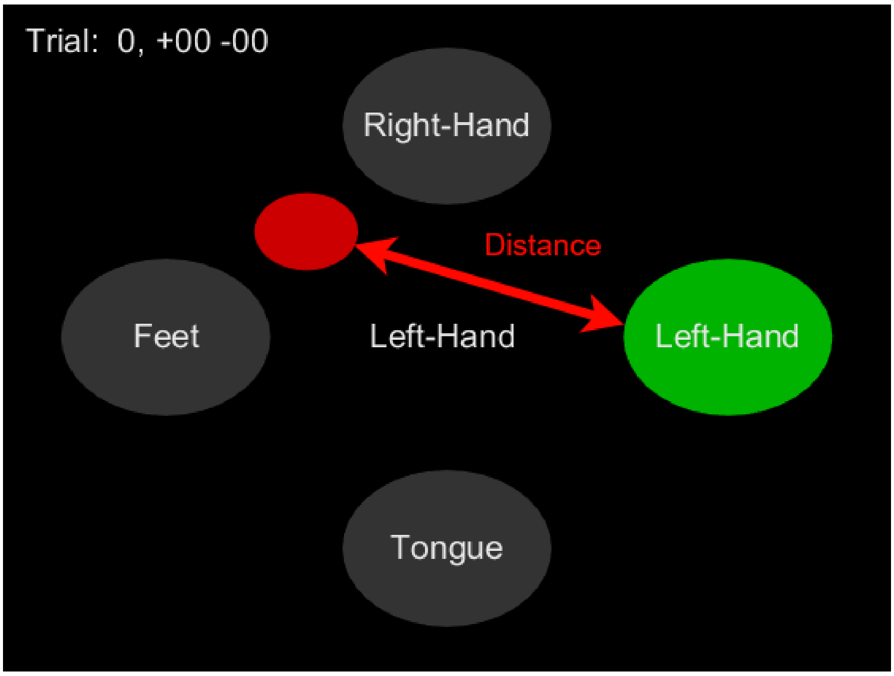

Figure 5: Visualisation of reward calculation. Buffer BCI uses the cursor (the red ellipse) to show its current prediction. Just like a user judges their own performance by the distance between the cursor and the task ellipse (in this case Left-Hand), the agent receives a reward based on this same distance: the euclidean distance between the cursor and task ellipse is calculated, and the negated value is returned as reward. This means that the larger the distance between cursor and task ellipse is, the lower the reward is.

noise of each task. This gaussian noise has a mean of 0, which means that the average noise over a large amount of states will be zero. A common technique to extract ERPs from an EEG signal is to average over a large amount of epochs (recording periods) time-locked to events [40]. Averaged over a large amount of states, the noise for each task will also be zero. This makes for noise that is similar to real-life noise that is present in EEG. Incidentally, the states in the model also represent epochs, although slightly different ones. This well be expanded upon in a later paragraph.

2.1.3 Rewards

The Buffer BCI system shows the user its current prediction using the cursor. The cursor being closer to the task ellipse means the user is performing well, since the system predicts the user is performing the correct task. The model uses the exact same concept. From the total output of a state (i.e. the activity + noise for each task) a prediction is generated. This prediction leads to a cursor position, and the distance between the task ellipse and the cursor is used as a negative reward. This means the lower the distance between the cursor and the task ellipse, the higher the reward. This is exactly how the Buffer BCI gives feedback to a user. In a Buffer BCI continuous feedback trial, the user constantly receives feedback. In a continuous trial of the model, the agent receives a reward for each state visited. In a Buffer BCI discrete feedback trial, the user only receives feedback at the end of a trial. In a discrete trial of the model, the agent only receives feedback in the last state of the trial. At the end of a trial, the distance between the predicted task ellipse and the trial task ellipse is used as a reward. This is both for the continuous and discrete trials, since the feedback phase (as seen in Figure 1) also happens in a continuous trial. In discrete trials, where only the last state returns a reward, the activity and noise of all the states is averaged, and based on this a prediction is made. This means that the prediction is based on the brain activity over the entire trial, which is also how the Buffer BCI predictions work.

The calculation of the reward using the distance between the cursor ellipse and the trial task ellipse is equal to the one in the Buffer BCI system. The position of the cursor relative to the task ellipse is calculated. The euclidean distance between the trial task ellipse and the cursor ellipse is then calculated, and the negated value of the distance is returned as the reward. At the last state of a trial, the closest task ellipse to the cursor is predicted, and the distance between the predicted task ellipse and trial task ellipse is used as reward. The exact calculation used to calculate the reward is explained in appendix A. For a visualisation of the reward calculation, see Figure 5.

Optimistic and pessimistic feedback is given by sampling a random gaussian value multiplied with the *reward noise* parameter and adding this to the true feedback. The gaussian value ensures that all feedback is not raised by a static amount. For optimistic rewards, only positive noise gets added to the rewards. For pessimistic rewards, only negative noise is added. This method of drawing a sample from a Gaussian distribution is also used by Barbero and Grosse-Wentrup to give biased feedback [38].

2.2 Q-Learning

The reinforcement learning algorithm chosen to investigate the effects of the different parameters is Q-Learning. Q-learning is one of the simplest reinforcement learning algorithms, yet is surprisingly powerful [44]. It uses a learning rule that has been shown to be able to predict choices in animals [34]. Furthermore, since the goal of this research is to investigate the learning differences using different neurofeedback parameters, it is not necessary to use a very advanced algorithm that may be more efficient. Q-learning uses Q-states, which store a value (called Q-values) for each state-action pair that has been encountered. Q-values are updated using the following update rule:

$$Q(s_t, a_t) \leftarrow \underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \times \left(\overbrace{\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{estimate of optimal future value}}}^{\text{learned value}} - \underbrace{Q(s_t, a_t)}_{\text{old value}} \right)$$

Often, reinforcement learning algorithms are used in environments with a finite amount of states. Algorithms keep learning until the environment has reached a terminal state [31]. Such a subsequence, from a starting state to a terminal state, is called an episode. After the agent has reached a terminal state, the environment is reset, and the agent starts again from a starting state. For this model, the logical choice for episodes was trials, since trials are the time-bound subsequences of which a learning session consists. The goal of one trial is to improve the user's skill on one specific task. So one episode represents a single Buffer BCI trial. Such a trial consists of multiple states, which we call *epochs*. One epoch represents one brain measurement step. The amount of epochs in a trial and the modulation rate can be set in such a way that a trial represents a real Buffer BCI trial of several seconds. The starting states consist of a a random task, which doesn't change during the trial, and zero task activity and noise.

The values that are kept by the states (i.e. task activity and noise) are continuous, which means the amount of possible states that could be encountered is infinite. Since this allows for a too high space complexity, the states are discretised when stored as Q-states. The task activity and noise are summed and then rounded to one decimal place. This limits the complexity to a level that can still be run on a personal computer. This discretisation is done by the Q-learning algorithm, but has no effect on the model itself. The states itself still store continuous values, and calculating the reward is also based on these continuous values. This keeps the learning, and its limitations, separated from the model itself.

There is no real solution to the exploration vs. exploitation dilemma [31]. Suggested by Sutton, and the strategy most widely used in the literature, is the ϵ-greedy strategy [45]. The ϵ-greedy strategy makes use of the ϵ parameter. This parameter determines how often the agent explores (i.e. chooses a random action) versus exploiting known knowledge. If this parameter is set to 50%, every other action will be random. Since this method is simple and effective, and there are no obvious better alternatives, this method is used by the Q-learning agents in the experiments.

2.3 Parameters

This paragraph states the parameter settings used in the experiments. Four tasks are used in the experiments, as all the research done by the group uses four tasks. These tasks can be named (e.g. left hand IM, right hand IM, etc.), but basically represent four arbitrary tasks that induce ERDs in the brain of the user. The value of the modulation rate determines the complexity of the model. The lower the modulation rate, the higher the complexity of the model since there will be more discrete steps to get to full activity in a task. For these experiments it has been set to 0.25. This means that there are four actions needed to get to full activity in one task. Pfurtscheller and Neuper (2001) state that it takes 250-500ms from cue-onset to discriminate between right and left hand imagined movement [46]. Using this information, and figure 2, we make the assumption that it takes 0.25 seconds to go from zero activity to full activity in one task. Since a trial is meant to represent a real BCI trial of 5 seconds (see Figure 1), to have one trial represent 5 seconds it consists of $5 \div 0.25 \div 0.25 = 80$ epochs. dvCalFactor has been set to 6, which for task activity of one task at 100% activity and three tasks at 0% will return a prediction of 99% for the task that is being performed.

There are several Q-Learning parameters that normally have to be set in a way that optimises learning in a certain environment. Often, these are set using trial-and-error [31]. However, since it's not needed to optimise the learning in this research, these parameters do not have to be fine-tuned. The learning rate α, determines how valuable new information is to the learner [31]. If α is set to a high value, recent rewards weigh more heavily than long-past ones, and newly acquired information will override old information. The learning rate α has been set to the standard value of 0.1 [31]. The discount factor γ, determines the importance of future rewards. A high discount factor will make the agent strive for long-term rewards, since the rewards of the possible future states will weigh more heavily. The discount factor γ has been set to 0.9. The last parameter that needs to be set is the exploration rate, ϵ. The exploration rate determines how often the agent will explore a random action to see if that action might be more profitable than ones already encountered [45]. A low exploration rate will lead to higher short-term rewards, but might be bad in the long-term since actions that may be more valuable are not explored. The exploration rate ϵ has been set to 0.1, one out of every ten actions a random action is performed.

2.4 Analysis

After every episode (i.e. trial), the policy generated by the learner is evaluated. The policy is a mapping of a state to the action with the highest Q-value in that state [31]. The policy is evaluated on all four tasks by performing the action returned by the policy for each state encountered. If the policy does not contain a certain state, a random action will be performed. The cumulative reward over these four trials is the evaluation score. The number of trials to run for an experiment is determined by hand, by doing a number of test runs to see how many trials are needed before the learning rate converges (i.e. the agent has approximately maximised his learning). Since there is an element of randomness (i.e. the selection of actions) to the Q-learning agent, the experiment will be run 60 times, to ensure that meaningful results are obtained. Each agent represents a subject, and groups of subjects with different model parameters will be compared. The metric

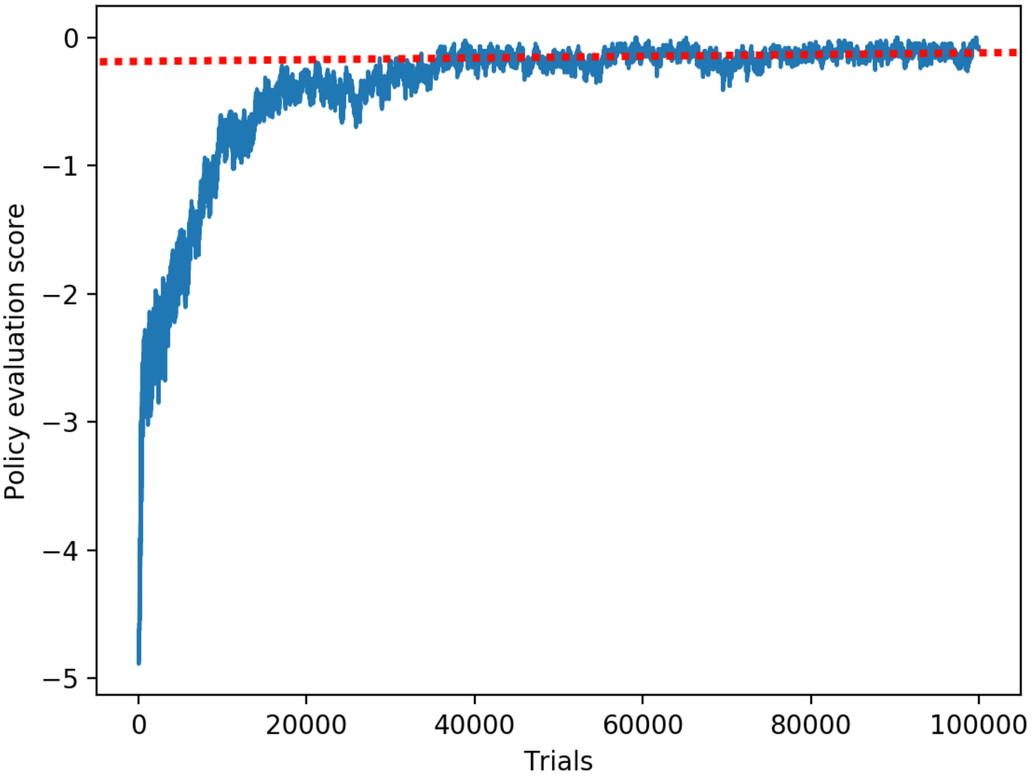

Figure 6: Visualisation of the convergence threshold. The amount of trials it takes an agent to cross the convergence threshold is the measure used to compare conditions: the trials-to-convergence.

that is used to compare different groups of subjects is called the *trials-to-convergence*. The lower the trials-to-convergence is, the faster the agent will have learned.

Sixty agents are trained, and after every trial their policy is evaluated, which results in a score for every trial. The scores for these sixty agents are then averaged. The score over the last 1000 trials (when the learning rate has converged) is averaged, which results in the average convergence score. 95% of this score is used as the *convergence threshold*, to make sure that each agent reaches it. The convergence threshold is the threshold by which each agent has approximately maximised his learning, see Figure 6 for an example of the convergence threshold. For each agent the trials-to-convergence measure is then calculated, by counting the amount of trials it takes for the agent to cross the convergence threshold. This results in a single measure of learning for each agent: the number of trials to convergence. But the value of the convergence threshold is also important, since a very high amount of noise may lead to agents converging at a very low level of performance. The value of the convergence threshold will also be reported, and we call this value the *performance-at-convergence*.

Theoretically, given enough time, every agent should be able to obtain optimal performance. Since noise is gaussian, after enough trials the noise should have canceled out and the real optimal actions will be known to the agent. However, as can be seen in Figure 6, it takes a lot of trials for the noise to completely cancel out, at 100,000 trials the performance still fluctuates. Yet, the agent in the figure has approximately converged, performance is very close to optimal and is relatively stable. The optimal performance-at-convergence for continuous feedback equals -6, but will be reported as 0 (i.e. 6 will be added to all performance-at-convergence measures), so that performance can be more easily compared. The optimal performance-at-convergence for discrete feedback is 0.

To investigate the effects of continuous and discrete feedback on learning, a Student's t-tests will be used if the data conforms to the following requirements: 1) the data follows a normal distribution, 2) the two samples have equal variances [47]. Normality of the data will be tested using the Shapiro-Wilk test, which has been shown to be the most powerful normality test [48]. Equivalence of variances will be tested using Levene's test for equality of variances [49]. If the data is normal but the two groups do not have equal variances, Welch's t-test will be used, which is more reliable for samples with unequal variances [50]. If the data turns out to not be normally distributed, the non-parametric Mann-Whitney U test will be used. They will be compared while keeping other variables static, so the continuous and discrete groups will be compared without any noise.

Since different levels of optimism and pessimism in the feedback might have different effects, experiments will be run with increasing amounts of optimism and pessimism (starting with 0.01, and repeatedly multplying by 2). The means for these different levels will be compared to see which level of optimism and pessimism has the most effect, using a boxplot. To see if optimistic and pessimistic reward noise have an effect, the factor of noise to compare is selected by plotting the different levels and choosing the trials-to-convergence mean that is the lowest of the different levels. This level will then be compared to the neutral condition for both the optimistic and pessimistic feedback conditions using either a Student's t-test or Mann-Whitney U test, depending on the requirements mentioned above.

To investigate the effect of noise on learning, we are interested in the correlation between the signal-to-noise ratio (i.e. the amount of system noise) and the rate of learning (i.e. trials-to-convergence). Increasing levels of noise will be investigated, starting at 0.001 and multiplying this by 2 repeatedly. The data will first be tested for normality using the Shapiro-Wilk test. The data will also be screened graphically to determine how the speed of learning and amount of noise are related, and different curves will be fit to determine what relationship describes the data best. Depending on the outcome of the screening, either Pearson's correlation coefficient (for a possible linear relationship) or Spearman's rank correlation coefficient (for a non-linear relationship) will be calculated. Since Spearman's rank-order correlation is non-parametric, normality is not a

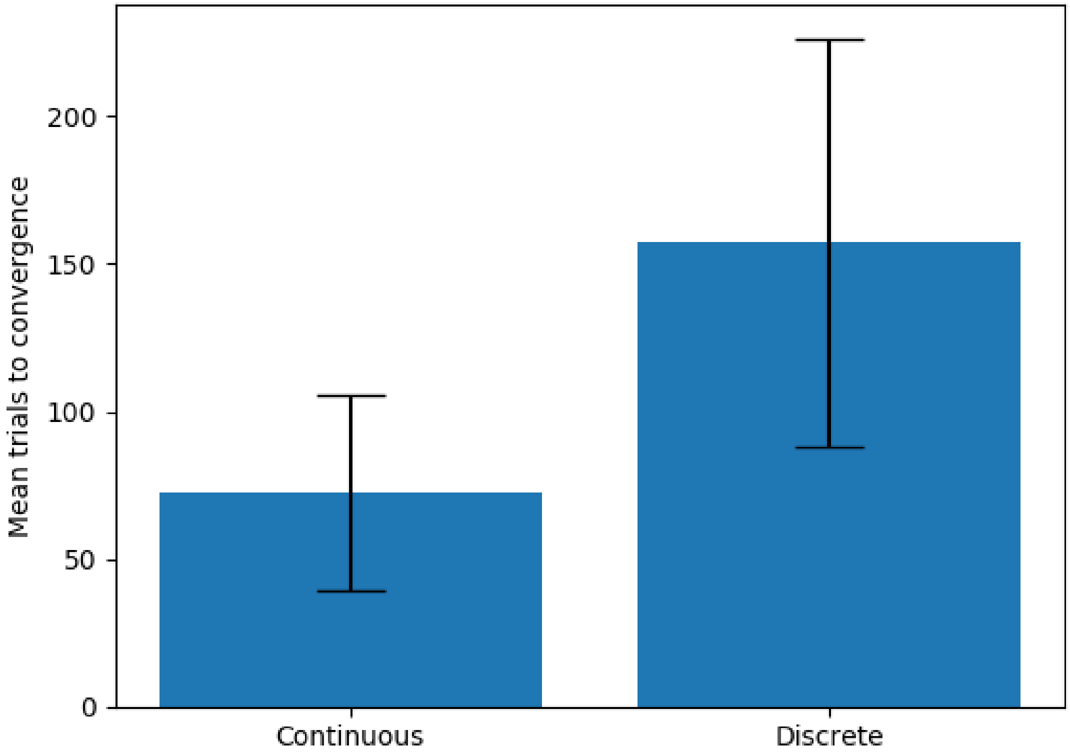

Figure 7: Graph of the continuous and discrete feedback conditions. The height of the bars represents the mean trials to convergence for each condition, and the error bars depict the standard deviation for each condition. The two conditions are compared with system noise set to 0.

requirement for the data, whereas it is a requirement for Pearson's correlation coefficient [51]. The results of the curve fitting will also be reported to illustrate the relationship between the two variables. To investigate a possible interaction between noise and continuous and discrete feedback, a two-way ANOVA will be used. ANOVA has the same assumptions as the Student's t-test, but since this question cannot be empirically investigated, ANOVA will be used regardless of the shape of the data. Ellis (2013) states that if one of the assumptions is violated, the p-value might not be trustworthy anymore [52], but in this case we can accept this. We are interested in a possible interaction effect.

All code used in the experiments and analysis can be found at
https://gitlab.socsci.ru.nl/d.roos/bcirflearning

3 Results

3.1 Continuous vs. discrete feedback

We compare the trials-to-convergence for continuous and discrete feedback with zero system noise and zero reward noise. After removal of outliers, both the continuous and discrete feedback conditions turn out to be normally distributed (Shapiro-Wilk df = 60, p = .075 and df = 56, p = 0.094 respectively). This means that depending on the variance, either a Student's t-test or Welch's t-test can be used. The variances of both groups can not be assumed to be equal (F = 13.473, p ¡ .001), so Welch's t-test will be used. The hypothesis is that learning will be faster in the continuous feedback condition than in the discrete feedback condition, so a one-tailed test will be performed ($H_1 : \mu_{\text{continuous}} <= \mu_{\text{discrete}}$). According to Welch's t-test, the mean of the continuous feedback condition is significantly lower than the mean of the discrete feedback condition (t(93.906) = -9.023, p < .001). See Figure 7 for a visualisation of the results for the two feedback conditions. The performance-at-convergence for the continuous condition is optimal, at 0, and for discrete feedback it is very close to optimal, at -0.03.

3.2 Optimistic and Pessimistic feedback

Since optimistic and pessimistic feedback might have different effects depending on the level of optimism or pessimism, we want to investigate what level is optimal to learning. We do this by boxplotting the different levels and choosing the level with the highest rate of learning (i.e. lowest average trials-to-convergence). At .04 optimistic reward noise we find the lowest average trials-to-convergence (M = 43.72, SD = 21.514), so this level of reward noise will be used to compare optimistic and pessimistic feedback with neutral feedback. System noise is set to 0 for the optimistic, pessimistic and neutral groups. It is hypothesized that both optimistic and pessimistic feedback will lead to slower learning than neutral feedback ($H_1 : \mu_{\text{optimistic}} <= \mu_{\text{neutral}}$, $H_2 : \mu_{\text{pessimistic}} <= \mu_{\text{neutral}}$). The neutral condition (i.e. normal feedback) is normally distributed (Shapiro-Wilk p > .05). The optimistic feedback condition is normally distributed after removal of outliers (Shapiro-Wilk p > .05), so a Student's t-test can be performed. Levene's test for equality of variances is significant (F = 18.208, p < .001), so equal variances cannot be assumed. According to Welch's t-test the mean of the optimistic condition is significantly lower than the mean of the neutral condition (t(96.725) = -5.987, p < .001). The pessimistic feedback condition is normally distributed (Shapiro-Wilk p > .05). Levene's test for equality of variances is insignificant (F = 3.034, p > .05) so equal variances can be assumed. The Student's t-test shows that the mean of the pessimistic feedback condition is significantly higher than the mean of the neutral condition (t(118) = 14.739, p < .001). See Figure 8 for a visualisation of the results for the three feedback conditions. The performance-at-convergence for both conditions is optimal (i.e. both are 0).

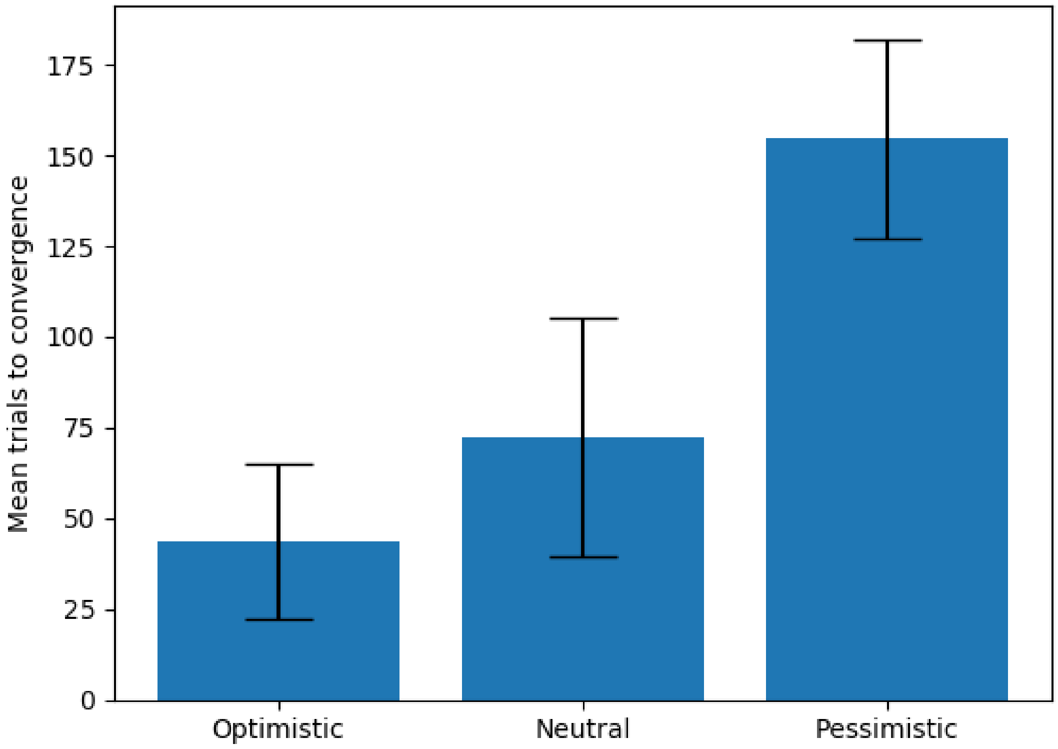

Figure 8: Graph of the optimistic, neutral, and pessimistic feedback conditions. The height of the bars represents the mean trials to convergence for each condition, and the error bars depict the standard deviation for each condition. The three conditions are compared with system noise set to 0.

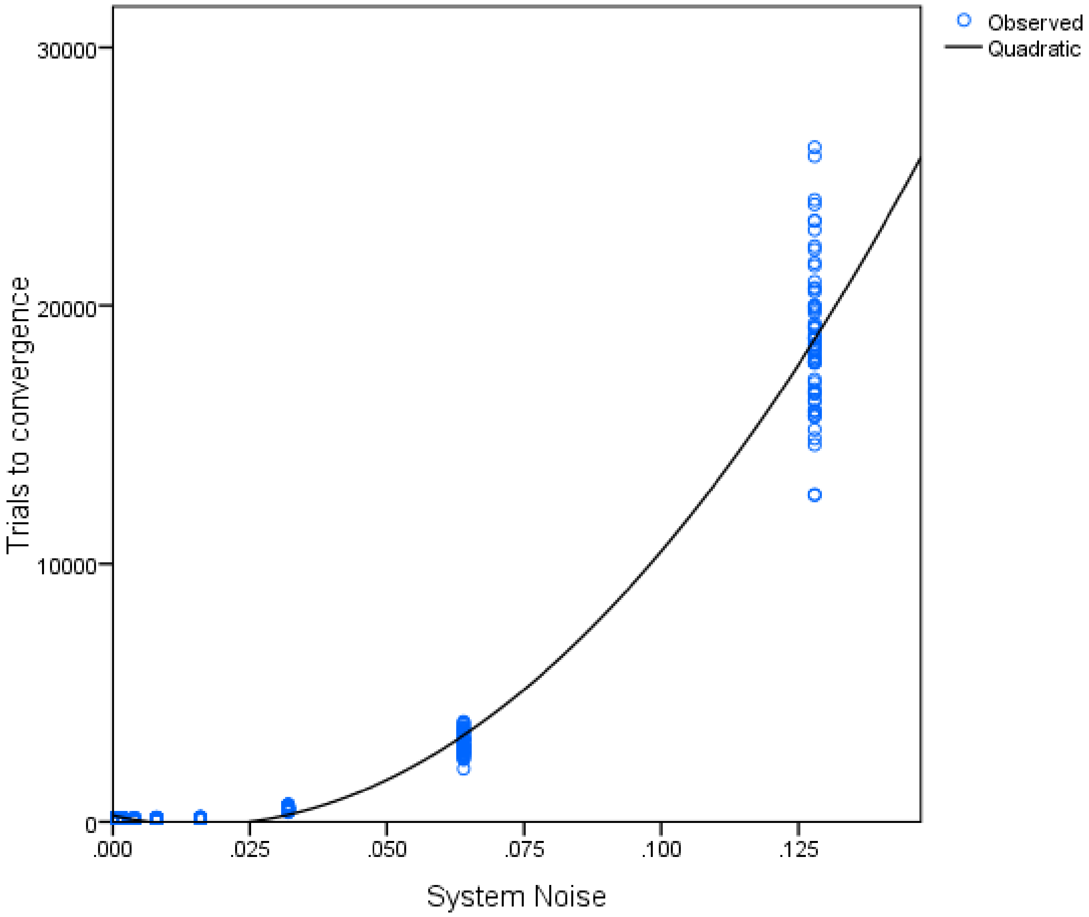

Figure 9: Graph of quadratic curve fitted to the continuous data with increasing amounts of system noise.

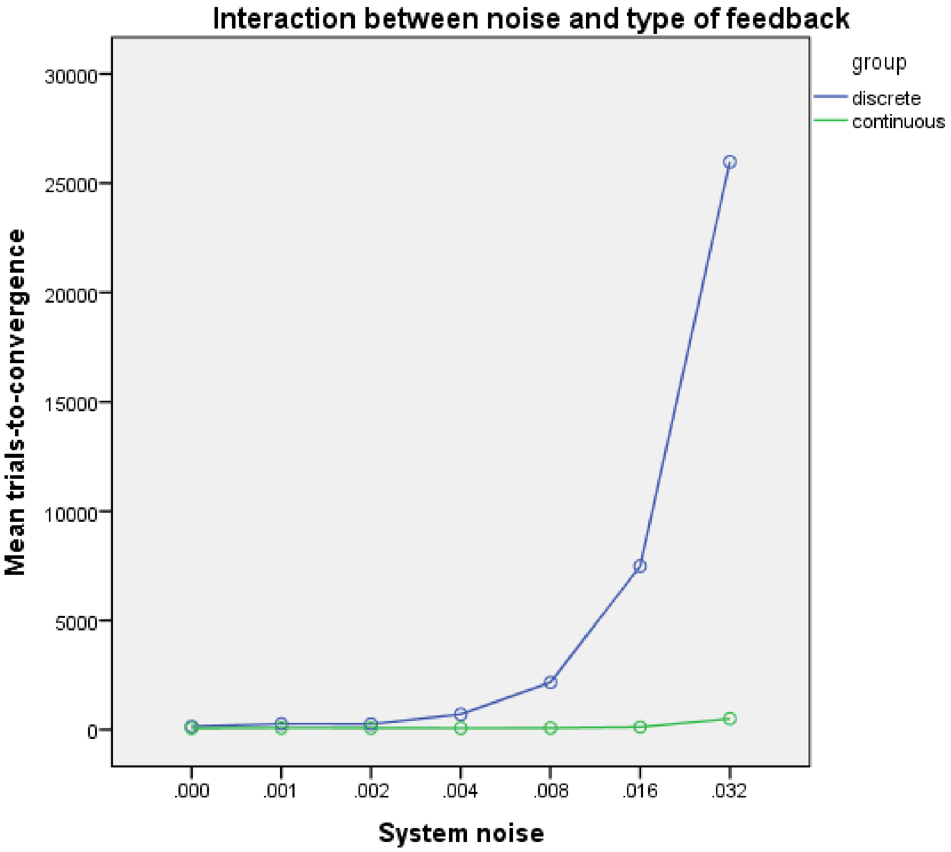

Figure 10: Plot of the interaction effect between type of feedback and amount of system noise. The effect of noise seems to get bigger for discrete feedback than continuous feedback as the noise increases.

3.3 The relationship between system noise and rate of learning

For the continuous data we investigate noise levels starting at 0.001 up to 0.128, in steps multiplied by 2 (i.e. 0.001, 0.002, 0.004 etc.). It is found that the data is not normally distributed (Shapiro-Wilk $p < 0.001$), so the Spearman's rank correlation coefficient will be used regardless of the type of relation between the variables. Using curve fitting we find that the data is very well described by a quadratic relationship (F(537) = 9393.295, $p < .001$, $R^2 = 0.972$). A visualisation of this quadratic curve fitted to the data can be seen in Figure 9. Since the data is best described by a non-linear relationship, we use Spearman's rank correlation coefficient to determine the correlation. We find a significant positive monotonic relationship between trials-to-convergence and system noise, as given by Spearman's rank correlation coefficient (N = 540, $\rho = .802$, $p < .001$). For discrete data we investigate noise levels starting at 0.001 up to 0.064, since noise levels higher than 0.064 take too long to run. We find that the data does not have a normal distribution (Shapiro-Wilk $p < 0.001$). So the Spearman's rank correlation coefficient will be used regardless of the type of relation between the variables. We find that this data also fits a quadratic relationship very well (F(477) = 5151.391, $p < .001$, $R^2 = 0.956$). There is a significant positive correlation between time-to-convergence and system noise as given by Spearman's rank correlation coefficient (N = 480, $\rho = .937$, $p < .001$).

If we express the noise as a noise-fraction (i.e. $noise/(noise + signal)$), a quadratic relationship still has the best fit (F(537) = 9054.314, $p < .001$, $R^2 = 0.971$). However, the effects of noise have only been measured up to the noise level of 0.128. When the amount of noise gets higher it may be possible that a linear relationship is a better fit, but to test this more data would be needed. For the data that is available, a quadratic relationship is almost a perfect fit. Up to the level of 0.064 system noise, the performance-at-convergence is optimal. At a system noise level of 0.128 the performance-at-convergence is -9, which is still close to optimal. But at this level of noise the performance still fluctuates at 80,000 trials, to get a better performance-at-convergence the agents would need to train for more trials. From the two-way ANOVA with type of feedback and amount of noise, we find a significant interaction effect between noise and type of feedback (F(6) = 818.453, $p < .001$, $eta^2 = .856$). See Figure 10 for a plot of the interaction effect.

4 Discussion

In this discussion I will try to answer the research questions using the results, existing literature, and the results of the other group members.

The results show that the continuous feedback condition has a significantly higher learning rate (i.e. lower average trials-to-convergence) than the discrete feedback condition. This is to be expected accounting for how Q-learning works, since feedback is slower which means that the agent will need more training to get to the same level of performance. Neuper et al. (1999) suggest that continuous feedback leads to more efficient BCI learning than delayed discrete feedback [36], while McFarland et al. (1998) show that continuous feedback can have either a facilitory or inhibitory effect depending on the learner. The literature is not yet sure about what kind of feedback leads to the most efficient BCI learning, but multiple factors might be in play. Since continuous feedback provides constant cursor movement, this might distract the user or lead to EEG responses that interfere with BCI control. Continuous feedback might decrease the signal-to-noise ratio, leading to worse performance. Especially in the beginning, when a user is not yet proficient in controlling a BCI and thus producing strong brain signals, this might work detrimental. Since motivation plays an important role in training [39], the most important thing for a new user might be to avoid demotivation. Discrete feedback may help with this aspect, since one can argue that this leads to a higher signal-to-noise ratio, and thus to better results for a new user. The results by Jordy Ripperda were insignificant due to too few test subjects, so there is no real point in comparing the model's results with his.

The results of the optimistic and pessimistic feedback experiments are surprising. Since both are essentially noisy rewards, it was hypothesized that both would lead to slower learning than the neutral condition. However, this turned out to be wrong. Optimistic feedback at a level of .04 (which had the lowest mean trials-to-convergence), turns out to be more efficient than the neutral condition. Barbero and Grosse-Wentrup (2010) hypothesize that biased feedback is especially helpful by increasing motivation when a user is not yet proficient at controlling a BCI. When a user is already capable, they found that biased feedback resulted in slower improvement. Pessimistically biased feedback has not been investigated, but one can argue that this would have a demotivating influence and thus lead to slower learning. The model's result show that pessimistic feedback at a level of .04 is slower than neutral feedback. It turns out that these results can be explained by how the model works. Because of the way predictions are calculated, even performing a correct action, the highest reward an agent can receive is negative. This means that if the agent performs a correct action for the first time, because of the ϵ-greedy strategy, all the other actions will be tried first, since they still have a Q-value of 0 while the Q-value of the correct action now has a slightly negative value. When giving optimistic rewards, the correct action will lead to a slightly positive Q-value. Because of greedy actions, this action will be performed from that point on, leading to faster learning. For pessimistic rewards, the correct action will be more negative than they are in the neutral feedback condition. This means that there's a higher chance that the correct action will lead to a lower reward than an incorrect action. If this happens, learning slows down, which explains the slower learning in the pessimistic feedback condition. The results from the pessimistic feedback condition follow the hypothesis, the negative rewards do not affect this condition. Negative rewards lead to exploration, since actions that have not been tried yet will always have a higher Q-value than actions that *have* been tried. The opposite happens when rewards are positive, where the action performed first will always be the greedy action until other actions have been performed. The odds of this happening depend on ϵ. So these results are explained by the use of negative rewards. The Q-learning algorithm still functions correctly, but the results of the optimistic feedback condition are a side-effect. One could expect that the hypothesis, optimistic and pessimistic feedback leading to slower learning, holds when the rewards are all positive.

Unsurprisingly, as the signal-to-noise ratio decreases, the learning rate also decreases. What is surprising, is that this relation follows an almost perfect quadratic equation. As the noise increases, the learning rate decreases quadratically. This means that low levels of noise are not that detrimental to learning, but as the noise increases, the effect of noise on learning will get bigger and bigger. So minimizing the signal noise should be one of the goals of BCI research, because it can have a big effect if the amount of noise is large enough. System noise seems to have a bigger effect on learning in the discrete feedback condition than the continuous condition. Possibly the effect of system noise also cancels out the rewards in the continuous condition, whereas in the discrete feedback any noise at the last state in the trial leads to an incorrect reward. Since the agent receives much less rewards in the discrete feedback condition, this might explain why the effect of noise is bigger in the discrete condition than in the continuous condition.

Although it is difficult to compare the results gained using the model with empirical results since research in this area is scarce, the model seems to reflect empirical data rather well. A lot of factors such as motivation or natural skill are not included in this model, yet the results are comparable to empirical data. I believe the current model reflects BCI learning relatively well, considering that it can represent different brain states (i.e. task activity), noise, and can be used to investigate different neurofeedback parameters such as continuous and discrete feedback and investigate the effect of noise to BCI learning. Although the model can still be improved, I believe this model provides a nice basis for further research, especially when the rewards are changed to be positive.

4.1 Future Work

Since the results for the optimistic and pessimistic feedback condition are a side-effect of the negative rewards, a better way of calculating rewards can be a future improvement. There's a good argument for the way rewards are calculated in the current model, so finding a way to calculate positive rewards that makes as much sense as these negative rewards may be difficult. But if done correctly, the optimistic and pessimistic feedback conditions can be investigated correctly.

Currently the model uses basic Q-states to store the value. This really limits the complexity of the model due to the amount of states that can be stored. It also does not reflect how a human brain learns skills, since a human is able to generalize its knowledge, which an agent is not able to do using a tabular method of storing Q-values. A better alternative would be function approximation, which allows the agent to generalize its stored knowledge [31]. Neural networks may be a perfect form of function approximation for this model, since they are based on neurons in the human brain [53]. Using such a form of function approximation would also allow the use of a continuous Markov Decision Process [31]. This could be an additional improvement to the current model, since the human brain does not run in discrete steps and this would increase the complexity of the model to be closer to the complexity of the human brain.

Additionally, a form of motivation could be added. As mentioned above, motivation plays a large role in skill learning in humans. The reason optimistically biased feedback improves learning in beginning users may be explained largely due to motivation. To simulate these effects motivation should be added to the model, leading to results that better reflect empirical data. Anoter possible improvement is separating the tasks from multiple actions. Neuper et al. (2005) show that the BCI performance of subjects using imagined movement is very reliant on the way they perform these tasks [54]. Kinesthetic motor imagery, where a user imagines actually moving a limb, performs much better than visual-motor imagery, where a user imagines viewing theirself in 3rd person performing the movement. This means that there are different ways of performing a task that have an effect on BCI performance. This could be included in the model, by having different actions modulate the same task at different levels. This would increase learning complexity, but also be closer to how humans learn BCI control.

4.2 Conclusion

A computational model of BCI learning was constructed, and reinforcement learning was used to train agents on the model. The learning rates of these agents were then compared with empirical data. The results reflect empirical data surprisingly well. In the model, continuous feedback leads to more efficient learning than discrete feedback, although the literature is not yet decided on this topic. Continuous feedback may lead to a lower signal-to-noise ratio, which at higher levels is very detrimental to learning, as shown by the model. The model shows that learning is more efficient using optimistic feedback than neutral feedback, and neutral feedback is more efficient than pessimistic feedback. Unfortunately, the results for the optimistic feedback condition turn out to be a consequence of the way rewards are calculated, so the question if optimistic feedback leads to more efficient learning than neutral feedback cannot be answered using the current model. Pessimistic feedback does seem to lead to slower learning, which was expected. To optimize learning, the model showed that noise should be minimized. As the noise grows, the learning slows down quadratically, so high levels are noise are very detrimental to performance. All in all, although the model can still be improved, it should provide a good basis for additional research.

References

[1] Jonathan R. Wolpaw, Niels Birbaumer, Dennis J. McFarland, Gert Pfurtscheller, and Theresa M. Vaughan. Brain-computer interfaces for communication and control. *Clinical Neurophysiology*, 113:767–791, 2002.

[2] Jonathan R. Wolpaw and Dennis J. McFarland. Control of a two-dimensional movement signal by a noninvasive brain-computer interface in humans. *Proceedings of the National Academy of Sciences of the United States of America*, 101(51):17849–17854, 2004.

[3] Jan van Erp, Fabien Lotte, and Michael Tangermann. Brain-computer interfaces: beyond medical applications. *Computer*, 45(4):26–34, 2012.

[4] Fabien Lotte and Camille Jeunet. Towards improved BCI based on human learning principles. In *Brain-Computer Interface (BCI), 2015 3rd International Winter Conference on*, pages 1–4. IEEE, 2015.

[5] Fabien Lotte, Florian Larrue, and Christian Mühl. Flaws in current human training protocols for spontaneous Brain-Computer Interfaces: lessons learned from instructional design. *Frontiers in Human Neuroscience*, 7, 2013.

[6] M. Barry Sterman and Tobias Egner. Foundation and Practice of Neurofeedback for the Treatment of Epilepsy. *Applied Psychophysiology and Biofeedback*, 31(1):21–35, March 2006.

[7] Moritz Grosse-Wentrup, Donatella Mattia, and Karim Oweiss. Using brain–computer interfaces to induce neural plasticity and restore function. *Journal of Neural Engineering*, 8(2):025004, April 2011.

[8] Anton Nijholt, Danny Plass-Oude Bos, and Boris Reuderink. Turning shortcomings into challenges: Brain–computer interfaces for games. *Entertainment Computing*, 1(2):85–94, April 2009.

[9] Luis Fernando Nicolas-Alonso and Jaime Gomez-Gil. Brain Computer Interfaces, a Review. *Sensors*, 12(12):1211–1279, January 2012.

[10] Stephan Waldert. Invasive vs. Non-Invasive Neuronal Signals for Brain-Machine Interfaces: Will One Prevail? *Frontiers in Neuroscience*, 10, June 2016.

[11] Jürgen Mellinger, Gerwin Schalk, Christoph Braun, Hubert Preissl, Wolfgang Rosenstiel, Niels Birbaumer, and Andrea Kübler. An MEG-based brain–computer interface (BCI). *NeuroImage*, 36(3):581–593, July 2007.

[12] Ranganatha Sitaram, Andrea Caria, Ralf Veit, Tilman Gaber, Giuseppina Rota, Andrea Kuebler, and Niels Birbaumer. fMRI Brain-Computer Interface: A Tool for Neuroscientific Research and Treatment. *Computational Intelligence and Neuroscience*, 2007:1–10, 2007.

[13] Noman Naseer and Keum-Shik Hong. fNIRS-based brain-computer interfaces: a review. *Frontiers in Human Neuroscience*, 9, January 2015.

[14] Sylvain Baillet, John C. Mosher, and Richard M. Leahy. Electromagnetic brain mapping. *IEEE Signal processing magazine*, 18(6):14–30, 2001.

[15] Marcel van Gerven, Jason Farquhar, Rebecca Schaefer, Rutger Vlek, Jeroen Geuze, Anton Nijholt, Nick Ramsey, Pim Haselager, Louis Vuurpijl, Stan Gielen, and Peter Desain. The brain-computer interface cycle. *Journal of Neural Engineering*, 6(4), August 2009.

[16] Gert Pfurtscheller and FH Lopes Da Silva. Event-related EEG/MEG synchronization and desynchronization: basic principles. *Clinical neurophysiology*, 110(11):1842–1857, 1999.

[17] G. Pfurtscheller, C. Brunner, A. Schlögl, and F.H. Lopes da Silva. Mu rhythm (de)synchronization and EEG single-trial classification of different motor imagery tasks. *NeuroImage*, 31(1):153–159, May 2006.

[18] E. Curran, P. Sykacek, M. Stokes, S.J. Roberts, W. Penny, I. Johnsrude, and A.M. Owen. Cognitive Tasks for Driving a Brain-Computer Interfacing System: A Pilot Study. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 12(1):48–54, March 2004.

[19] G. Pfurtscheller, C. Neuper, D. Flotzinger, and M. Pregenzer. EEG-based discrimination between imagination of right and left hand movement. *Electroencephalography and Clinical Neurophysiology*, 103(6):642–651, December 1997.

[20] Elisabeth V.C. Friedrich, Reinhold Scherer, and Christa Neuper. The effect of distinct mental strategies on classification performance for brain–computer interfaces. *International Journal of Psychophysiology*, 84(1):86–94, April 2012.

[21] Elisabeth V.C. Friedrich, Reinhold Scherer, and Christa Neuper. Stability of event-related (de-) synchronization during brain–computer interface-relevant mental tasks. *Clinical Neurophysiology*, 124(1):61–69, January 2013.

[22] E Curran. Learning to control brain activity: A review of the production and control of EEG components for driving brain–computer interface (BCI) systems. *Brain and Cognition*, 51(3):326–336, April 2003.

[23] Elisabeth V.C. Friedrich, Reinhold Scherer, and Christa Neuper. Long-term evaluation of a 4-class imagery-based brain–computer interface. *Clinical Neurophysiology*, 124(5):916–927, May 2013.

[24] Jens Kohlmorgen, Guido Dornhege, Mikio Braun, Benjamin Blankertz, Klaus-Robert Müller, Gabriel Curio, Konrad Hagemann, Andreas Bruns, Michael Schrauf, Wilhelm Kincses, and others. Improving human performance in a real operating environment through real-time mental workload detection. *Toward Brain-Computer Interfacing*, pages 409–422, 2007.

[25] J. Kamiya. Conscious control of brain waves. *Psychology Today*, 1:56–60, 1968.

[26] H. Marzbani, H. Marateb, and M. Mansourian. Methodological Note: Neurofeedback: A Comprehensive Review on System Design, Methodology and Clinical Applications. *Basic and Clinical Neuroscience Journal*, 7(2), 2016.

[27] David J. Vernon. Can Neurofeedback Training Enhance Performance? An Evaluation of the Evidence with Implications for Future Research. *Applied Psychophysiology and Biofeedback*, 30(4):347–364, December 2005.

[28] Mirjam EJ Kouijzer, Jan MH de Moor, Berrie JL Gerrits, Marco Congedo, and Hein T. van Schie. Neurofeedback improves executive functioning in children with autism spectrum disorders. *Research in Autism Spectrum Disorders*, 3(1):145–162, 2009.

[29] Tanju Sürmeli, Ayben Ertem, Emin Eralp, and Ismet H. Kos. Schizophrenia and the efficacy of qEEG-guided neurofeedback treatment: a clinical case series. *Neuroscience Letters*, 500:e16–e17, 2011.

[30] Christa Neuper and Gert Pfurtscheller. *Neurofeedback Training for BCI Control*, pages 65–78. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[31] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction.* Cambridge, MA: MIT Press, 1998.

[32] Abhijit Gosavi. Reinforcement learning: A tutorial survey and recent advances. *INFORMS Journal on Computing*, 21(2):178–192, 2009.

[33] Richard Bellman. A markovian decision process. *Indiana Univ. Math. J.*, 6:679–684, 1957.

[34] Matthew F S Rushworth and Timothy E J Behrens. Choice, uncertainty and value in prefrontal and cingulate cortex. *Nature Neuroscience*, 11(4):389–397, April 2008.

[35] A. GuruvaReddy and S. Narava. Artifact Removal from EEG Signals. *International Journal of Computer Applications*, 77(13):17–19, September 2013.

[36] C. Neuper, A. Schlogl, and G. Pfurtscheller. Enhancement of Left-Right Sensorimotor EEG Differences During Feedback-Regulated Motor Imagery. *Journal of Clinical Neurophysiology*, 16(4):373–382, July 1999.

[37] Dennis J. McFarland, Lynn M. McCane, and Jonathan R. Wolpaw. EEG-based communication and control: short-term role of feedback. *IEEE Transactions on Rehabilitation Engineering*, 6(1):7–11, 1998.

[38] Álvaro Barbero and Moritz Grosse-Wentrup. Biased feedback in brain-computer interfaces. *Journal of neuroengineering and rehabilitation*, 7(1):34, 2010.

[39] Edward L. Deci. Effects of externally mediated rewards on intrinsic motivation. *Journal of personality and Social Psychology*, 18(1):105, 1971.

[40] Michal Teplan and others. Fundamentals of EEG measurement. *Measurement science review*, 2(2):1–11, 2002.

[41] Christa Neuper, Reinhold Scherer, Selina Wriessnegger, and Gert Pfurtscheller. Motor imagery and action observation: Modulation of sensorimotor brain rhythms during mental control of a brain–computer interface. *Clinical Neurophysiology*, 120(2):239–247, February 2009.

[42] Gert Pfurtscheller and Christa Neuper. Motor imagery activates primary sensorimotor area in humans. *Neuroscience letters*, 239(2):65–68, 1997.

[43] René Marois and Jason Ivanoff. Capacity limits of information processing in the brain. *Trends in Cognitive Sciences*, 9(6):296–305, June 2005.

[44] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

[45] Michel Tokic. Adaptive ϵ-greedy exploration in reinforcement learning based on value differences. In *Annual Conference on Artificial Intelligence*, pages 203–210. Springer, 2010.

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

[47] Carol A. Markowski and Edward P. Markowski. Conditions for the effectiveness of a preliminary test of variance. *The American Statistician*, 44(4):322–326, 1990.

[48] Nornadiah Mohd Razali, Yap Bee Wah, and others. Power comparisons of shapiro-wilk, kolmogorov-smirnov, lilliefors and anderson-darling tests. *Journal of statistical modeling and analytics*, 2(1):21–33, 2011.

[49] Morton B. Brown and Alan B. Forsythe. Robust Tests for the Equality of Variances. *Journal of the American Statistical Association*, 69(346):364, June 1974.

[50] Ben Derrick, Deirdre Toher, and Paul White. Why Welch's test is Type I error robust. *The Quantitative Methods in Psychology*, 12(1):30–38, 2016.

[51] Anthony J. Bishara and James B. Hittner. Testing the significance of a correlation with nonnormal data: Comparison of Pearson, Spearman, transformation, and resampling approaches. *Psychological Methods*, 17(3):399–417, 2012.

[52] J. Ellis. *Statistiek voor de Psychologie.* Boom Lemma, 2013.

[53] Judith E. Dayhoff and James M. DeLeo. Artificial neural networks. *Cancer*, 91(S8):1615–1635, 2001.

[54] Christa Neuper, Reinhold Scherer, Miriam Reiner, and Gert Pfurtscheller. Imagery of motor actions: Differential effects of kinesthetic and visual–motor mode of imagery in single-trial EEG. *Cognitive Brain Research*, 25(3):668–677, December 2005.

Appendices

A Reward calculation

The following calculation is used to create a prediction out of the task activities:

$$prob = exp((task\_activity - max(task\_activity) * dvCalFactor)$$

$$prob = prob/sum(prob)$$

The calculation of the reward using the distance between the cursor ellipse and the trial task ellipse is equal to the one in the Buffer BCI system. From every task output the maximum task output is subtracted, and is multiplied with the *dvCalFactor*. The dvCalFactor is a parameter used to make small changes in predictions lead to larger changes in the cursor position. If dvCalFactor would not be used, the user would hardly see any change in the cursor position, which would not be useful feedback. In the model dvCalFactor has been set in a way such that full activity in one task and zero activity in the different tasks, will lead to a prediction of $\tilde{9}9\%$ for the full activity task. After the values have been multiplied with the dvCalFactor, the exponential function is used on the probabilities. This scales the prediction, and ensures that even negative values (which can occur due to noise) will lead to a positive prediction. After this, the predictions get normalised and multiplied with the task ellipses. This results in the new cursor position. The euclidean distance between the trial task ellipse and the cursor ellipse is then calculated, and the negated value of the distance is returned as the reward. At the last state of a trial, the closest task ellipse to the cursor is predicted, and the distance between the predicted task ellipse and trial task ellipse is used as reward.