BCI Practical Course

A BCI to play Space Invaders

Group 9
Zepp Uibo (s1012287)
Marjolein Troost (s4381777)
Dennis Doerrich (s1001070)

February 8, 2018

1 Introduction

The aim of a Brain Computer Interface (BCI) is to decode human intention that is transmitted through brain signals, so without use of the peripheral nervous system, in order to control a computer. When the user actively tries to communicate his intentions to the computer, the BCI is called *active*. A *passive* BCI is where the user does not actively try to communicate, but his brain-states are simply monitored. In this report we will focus on the active BCI.

For an active BCI to work, we need a brain signal that the user can learn to change, a way of measuring that signal and a way of understanding the signal that is measured.

Active BCIs can be divided into roughly two groups: *evoked* and *induced*. Evoked BCIs use events that cause a certain response in the users brain signal that is time-locked to the event. The user can choose to attend to this event or not and in this way to change the brain signal. Evoked events are easy to detect since the brain response has an exact time-lag after the event, so averaging the signal over multiple events will show the brain response clearly.

Induced BCIs do not use use a time-locked signal but rather another feature of the signal that is time-locked, like a change in power at a certain frequency.

We will consider a BCI in which a game is controlled using (imagined) hand movements. The movement will cause a reduction in μ (8-13 Hz) and β (12-30 Hz) power over the hand region in the motor cortex (Cui et al. (1999), Beisteiner et al. (1995)), so we will be using an active, induced BCI.

Generally, the electrical signals emitted by neural activity in the brain are very small (around 10 μV). There is also a lot of noise in the signal that we measure, so to be able to isolate the brain signal, we need to maximize the signal-to-noise ratio. We can do this using preprocessing of the signal, e.g. selecting only the interesting frequencies.

Aside from the noise, the signal also varies a lot between subjects, sessions and even within sessions because of random noise. Therefore, usually a subject-dependent classifier is needed. This classifier is trained in a separate calibration session in which the user performs a known series of mental tasks.

In this report we have focused on three separate, but related questions.

1 Can we improve (imagined) movement BCI performance by choosing a one hand motor task that produces a large brain signal or by training the classifier on many different tasks?

2 Can we reduce the difficulty of train-test transfer by calibrating in a way that is most similar to the test?

3 Can we improve performance on the game by changing the control to a more physically realistic one?

We based these questions on the general opinion that the task that the user performs is irrelevant for the performance of the BCI. However, Curran and Stokes (2003) note that:

For example, factors such as memory, concentration, attention and ***difficulty of the task (cognitive load)*** *have been shown to affect the changes in EEG signals (Gevins, Smith, McEvoy, & Yu, 1997; Gevins et al., 1998; Holländer, Petche, Dimitrov, Filz, & Wenger, 1997; McEvoy, Smith, & Gevins, 1998; Smith, McEvoy, & Gevins, 1999).*

This latter difficulty has implications for BCI research where factors such as attention and cognitive load do not (to date) appear to have been seen as relevant variables. Some recent papers have shown evidence, for example, that alpha activity is affected by changes in task difficulty and practice (Gevins et al., 1998; McEvoy, Smith, & Gevins, 2000; Smith et al., 1999). So, among other things (see also Bartolic et al. (1999), Bauer and Gharabaghi (2015)), the difficulty of the task should affect the performance of the BCI.

Many papers focus on transfer learning, i.e. improving the performance of the BCI with as little calibration as possible. For examples, see Jayaram et al. (2016) or Lotte (2015). Most of them, however, focus on improving the signal processing part of the BCI, such that the trained classifier can be reused for other subjects or in other sessions. We will focus on adapting the training task such that the classifier can be reused in other session.

Wolpaw and McFarland (2004) also uses movement encoding instead of position encoding for 2D cursor control. They find that this works well, but their experimental setting is very different from our experiment, see for a more elaborate discussion Section 2.1.

2 Methods

Since we have three main questions, we also have three experiments. We will discuss each of them separately. First, however, we will describe the set-up that is used for all three experiments.

2.1 General Set-Up

We have a subject(Dennis) seated in front of a computer screen. The screen will be showing the calibration or game environment. The subject's brain-signal will be measured using a 10-electrode EEG. The electrodes will be placed as in figure 2, so around the hand regions of the motor cortex, see also Figure 1. We use a MOBITA that sends the EEG signal to the computer where it will be processed. The preprocessing consists of several steps. First the data will be detrended to remove the DC offset. Then, usually, bad channel removal follows. In our case we will not remove bad channels since we have only ten electrodes. A spatial filter (adaptive whitening) is applied to decorrelate channels and increase the power of the weak signals. Features are extracted using Welch's method. Then the relevant features are selected using a spectral filter. The spectral filter

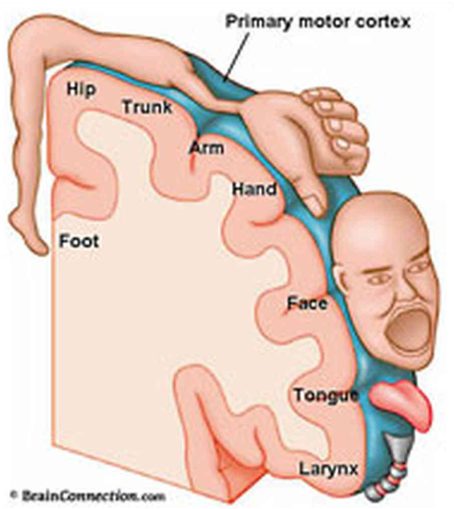

Figure 1: An artistic impression of the motor cortex. We can see that the hand regions are quite large. The electrodes on C3 and C4 should be directly over the hand regions.

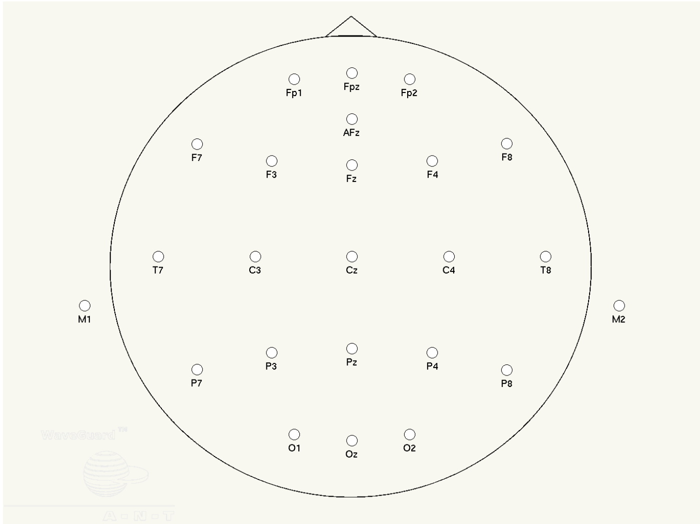

Figure 2: The electrodes are placed over F3, F4, T7, C3, Cz, C4, T8, P3 and P4. This is where we expect to find the strongest signal differences between the different tasks. See also Miller et al. (2010).

ramps up between 6 and 8 Hz, is constant between 8 and 28 Hz and ramps down between 28 and 30 Hz. Finally, we train a linear logistic classifier with quadratic regularization on the preprocessed data.

A calibration phase is used to train the classifier. In this phase the user has to perform a movement that is indicated on the screen. Indeed, we see that the classifier learns to distinguish between left and right with the signals from the right and left hemisphere. See also Figure 3.

The trained classifier will be used to control a cannon in the game *Brainfly*. Brainfly

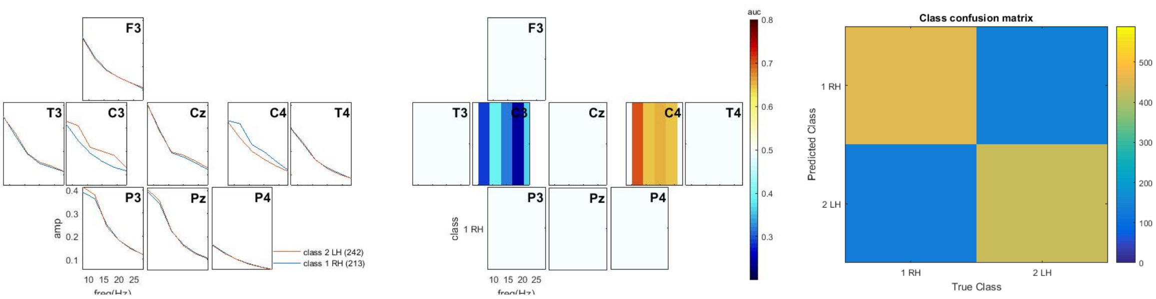

Figure 3: On the left we see the averaged frequency of the brain signal per electrode per hand. In C3 and C4 we clearly see a reduction in power in the μ and β range in the opposite hemisphere. The AUC image also shows an increase in power C4 and a decrease in power over C3 for the right hand movement. Note that one of the channels (F4) has been removed. The final image is the confusion matrix, showing how often the classifier predicts the correct or the wrong class.

is a simplified version of the game Space Invaders in which the user has to shoot aliens that come down from the top of the screen, increasing in size as they come down. The cannon can be controlled using brain signals. It will be shooting with a rate of maximally one shot per second. If the aliens are hit, the user gets points. The earlier the alien is hit, the more point the user gets for accuracy. If the alien is not hit before it touches the bottom of the screen, the user dies. An alien can only be hit, if it is the lowest alien on the screen. The game takes a total time of 90 seconds. Some screen shots of the game are shown in Figures 4, 5, and 6. In Wolpaw and McFarland (2004), a center-out design is used in combination with speed encoding. We will also use speed (or momentum) encoding, although our cannon is meant to change direction frequently which makes our situation very different from the situation in Wolpaw and McFarland (2004).

Figure 4: The cannon and an alien. The alien is still small.

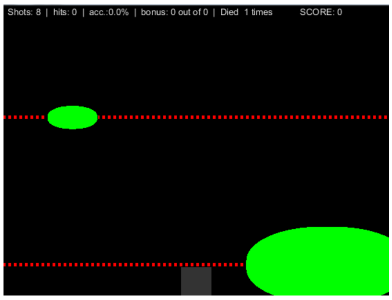

Figure 5: Two aliens on the screen. Shoot the lowest first.

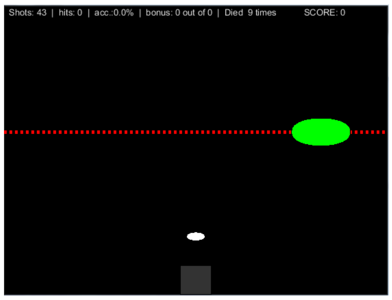

Figure 6: The cannon firing, while the alien is coming down.

Now, that we know the basic set-up, we can move on to our experiments.

2.2 Experiment 1

We selected five motor tasks of varying difficulty.

1 **Alternate:** move index and middle finger up and down alternatively as fast as possible.

2 **Pinch:** squeeze the tops of your thumb and index finger together and release.

3 **Piano:** resting your fingers on the table, pretend you play the piano.

4 **Cross:** try to cross your index and middle finger, first with the index finger on top, then with the middle finger on top. Alternate.

5 **Don't touch:** similar to pinch. Try to keep your thumb and index finger as close as possible without actually touching.

During all tasks the subject will look at the screen. On the screen the task will be indicated by its name. We chose to perform these tasks because of their diversity and their difficulty. It is hypothesized that more difficult tasks elicit a larger brain response;

therefore we expect a better performance for the more difficult tasks like *don't touch* (concentration) or *piano* (creativity). In any case, the diversity of the tasks will cause some training noise. We hope that this will make the classifier more robust and less sensitive to changes between sessions, subjects, or noise from the game. We expect the diversity of the tasks to improve the subject's performance on the game.

First we let the subject perform all of out five motor tasks separately on the standard implementation of the calibration. The resulting classifier prediction scores based on 40 trials per task are in Table 1. The subject also performs a calibration where the tasks are randomized. The task that the subject has to perform is shown in the middle of the screen on the fixation point between trials, see Figure 7. We do not use pseudo-randomized trials, so the number of trials per task is not perfectly balanced. Next, we let the subject in

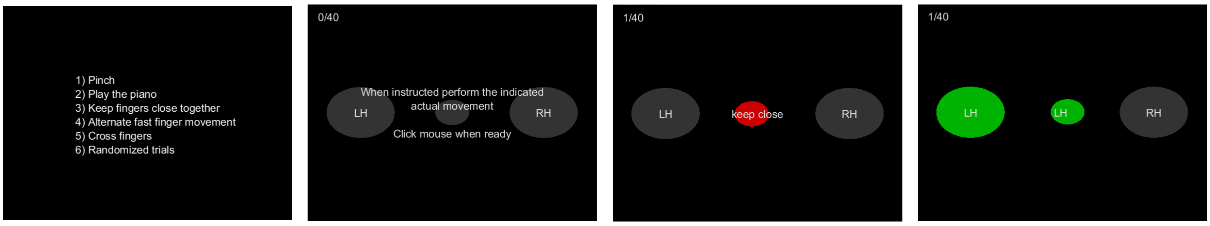

Figure 7: We see the menu, press 6 to select the random task, for example. The user is instructed to perform the indicated task. The task is indicated before the trial starts, in the red fixation point. The color of the oval on the left changes to green, indicating the task should be performed with the left hand in this case.

a single session perform the randomized version of the calibration five times, so for a total of 200 trials. We use these trials to extract data for each of the five tasks and train a classifier on each of the single tasks, see Table 2. Since we randomized the trials, the number of data is not the same for each task, but since we have a total of 200 trials, the numbers are sufficiently similar to be comparable. So after this task, we have six trained classifiers: one for each task and one trained on all randomized trials. Based on the cross-validation score, we pick the best of the five tasks and compare the classifier trained on that task with the classifier trained on 40 of the randomized trials. That way we can compare the benefits of the task giving the strongest brain signal with the benefits of having a more robust classifier. Comparing the performance will be done in a separate session so both classifiers suffer the same amount of train-test and session-session transfer problems.

We expect that when playing the game, the subject will get the highest scores using the classifier trained on the randomized trials because this classifier is more robust to changes in task. Because it is more robust to changes in task, we expect that the classifier can better handle the differences between sessions. When playing the game, the subject is free to control the cannon in any way he thinks is best. The results are in Table 3.

2.3 Experiment 2

The second experiment will use an adapted calibration setting that is more similar to the game than the standard calibration session, illustrated in Figure 8. Instead of the

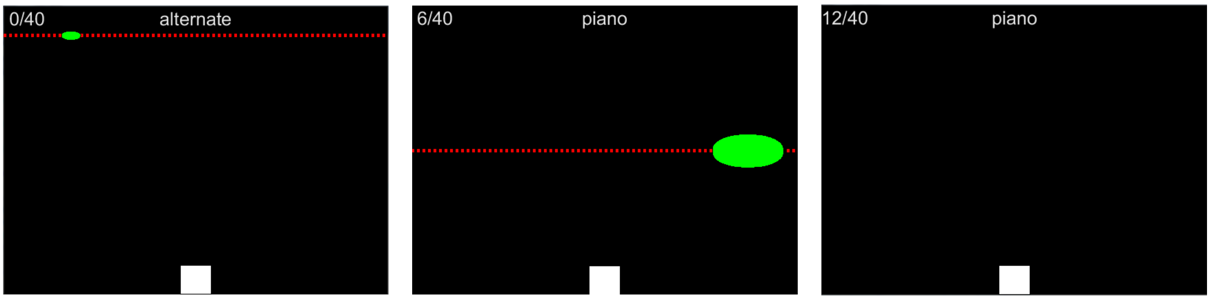

Figure 8: The task is shown at the top. Then an alien appears either on the left or on the right. Only one alien will be on the screen at the time. The task is shown before the alien appears and will remain on the screen while the alien is on the screen.

stationary green ovals that indicate with which hand to perform the indicated movement, we will have actual *aliens* (also green ovals) coming down on either side of the screen. If the alien appears left, the movement should be performed with the left hand. Which movement to perform will be shown at the top of the screen prior to and during the descent of the alien. In contrast to the game, there can never be two aliens on the screen at once and the cannon does not move or fire, so the aliens cannot be shot. We will train the classifier using the randomized tasks like in the first experiment, since more data usually gives a better performance. The trained classifier will be used to play four games in another session to ensure a fair comparison. The total score, accuracy and number of deaths will be compared with the classifier calibrated using the standard training environment on the randomized tasks. The results are in Table 4.

We expect the classifier trained in the noisy environment to be more robust to noise and therefore to produce better results when playing the game.

2.4 Experiment 3

The idea of the third experiment is comparing the performance on the game using the standard position encoding where the brain signal is linked to incrementing position in a certain (decoded) direction, with the performance on the game using momentum decoding where the brain signal is linked to incrementing speed in a certain (decoded) direction. We suspect that momentum decoding is physically more plausible and therefore easier to control by the brain. The speed of the cannon will be limited by a maximum, this simulates the presence of friction. When the cannon hits an edge, it's speed will be set to zero again. When the user wants to change direction, he will first have to reduce the cannon's speed in the current direction to zero, before being able to increase the speed of the cannon in the desired direction. In order to not getting stuck on either side of the screen, the velocity is set to zero, once the cannon is on its total left or right position.

We compare the performance on the game using the classifier trained on the random tasks. Whether we use the classifier that is calibrated in the normal environment, or the noisy environment, depends on their performance on the game. We will simply choose the best performing. The subjects will play four games with momentum and four without.

The total score, accuracy and number of deaths will be computed and compared. The results are in Table 5.

3 Results

Here we collected the tables with results from the experiments. In Tables 1, 2, and 3 we see the results from experiment 1. The results from the second experiment are in Table 4 and the results from the third experiment are in Table 5.

Looking at Table 1, we see that none of the percentages is significantly larger than the others (mean = 67.3%, standard deviation = 4.07%). From Table 1 we cannot conclude much about which task gives the best performance. An important note is that for the first four tasks, we see the score increase, so this could be a sequential effect. Even though we tried to remove the data that was collected before, we cannot guarantee that there is no learning or data pooling occurring here. The score for the piano task has been determined almost two months after the other scores.

Task description	Task name	Classifier Score
Put together thumb and index finger and squeeze lightly	Pinch	63%
Alternate index and middle finger swiftly	Alternate	69%
Cross index and middle finger, alternating top and bottom	Cross	70%
Hold thumb and index finger close without touching	Don't Touch	75%
Wiggle fingers like playing a piano	Piano	66%
All of the above in random order	Randomized	70%

Table 1: Description of each of the tasks and the short name used in the experiments. The classifier score is computed using cross-validation on a calibration of 40 trials.

Looking at Table 2 it is not obvious that these tasks are what lead to a total score of 70% since the slices get barely above 50%. There is one number much higher than the others, which is the value for piano, it is also not significant (mean = 55.86%, standard deviation = 6.21). Since the piano task stands out as the best in Table 2, and the results from Table 2 are more reliable because of the randomization, we pick piano as the best value.

With Tables 3, 4 and 5 we feel obliged to add that our dataset (normal calibration, 200 random trials) went missing, or was corrupted. This dataset had a classifier accuracy of 61.2% instead of the expected 70.6%. We also noticed that in this dataset the electrodes were probably placed wrongly on the EEG-cap. We suspect something or someone has corrupted the data or mixed up some of the data files. In Table 3: the piano task was sliced from this corrupt random data (240 trials, normal calibration). The classifier performance was 54% instead of the expected and previously found 65.6%. The random data was sliced from the random dataset with normal calibration earlier, so that uses the correct data. We are unsure how this happened. Because of this problem, we cannot draw conclusions from these data.

Task	Number of Trials	Classifier Score Random
Pinch	240	50.8%
Alternate	198	56.1%
Cross	216	50.0%
Don't Touch	312	52.0%
Piano	234	65.6%
Randomized	1200 = 5*240	60.68% average, 68.3% best

Table 2: For each task the number of trials (measuring 6 times per trial gives 6*40=240 measurements per task) and the performance of the classifier that is trained on these trials is given. Since the number of trials is variable, we need to use balanced loss. The classifier score *random* indicates the performance of a classifier trained on trials sliced from the 1200 random trials. The classifier score *separate* indicates the performance of a classifier trained on a separate calibration session for that task. In the last row the total number of trials and the performance of the classifier trained on all random trials is given. To compare the random classifier with the other classifiers, we sliced the data into five sequences and trained a classifier on each, then we averaged these performances giving the score 60.68%.

Game number	Type of Score	Classifier piano task	Classifier randomized task
1	accuracy	11.5%	12.6%
	score	64	71
	deaths	10	10
2	accuracy	12.8%	18.6%
	score	78	89
	deaths	11	5
3	accuracy	8.0%	24.4%
	score	47	136
	deaths	13	4
4	accuracy	11.6%	12.8%
	score	64	77
	deaths	10	11
Total	accuracy	43.9	68.4
	score	253	373
	deaths	44	30

Table 3: Comparing the classifier trained on a single task and the classifier trained on a set of 40 randomized trials. For each game the score, accuracy and number of deaths are given. In the last row an average performance is given for each of the classifiers.

In Table 5 we compare the game with position decoding and the game with momentum decoding. We found that when using momentum, the cannon was stuck in one corner all the time. The user could make it vibrate, so moving left and right in turn, but not move. Therefore, we abandoned the momentum control after two tries. We think that it would be possible to use momentum if the user is already very proficient at controlling the cannon, and even a higher score might be obtained in that case because the speed of the cannon will be increased if the user's intention is clear.

Game number	Type of Score	Normal calibration	Noisy calibration
1	accuracy	14.0%	17.4%
	score	65	102
	deaths	8	8
2	accuracy	17.4%	14.0%
	score	76	77
	deaths	5	10
3	accuracy	12.8%	19.8%
	score	59	104
	deaths	9	5
4	accuracy	11.5%	19.5%
	score	58	85
	deaths	10	3
Total	accuracy	55.7	70.7
	score	258	368
	deaths	32	26

Table 4: Comparing the classifier trained in the normal calibration environment and the classifier trained on the noisy calibration environment using 200 random trials. For each game the score, accuracy and number of deaths are given. In the last row an average performance is given for each of the classifiers.

Game number	Type of Score	Normal calibration	Noisy calibration
1	accuracy	11.6%	12.6%
	score	71	71
	deaths	11	10
2	accuracy	10.5%	18.6%
	score	54	89
	deaths	10	5
Total	accuracy	22.1	31.2
	score	125	160
	deaths	21	15

Table 5: Comparing the classifier performance on the game with and without momentum using 40 random trials. For each game the score, accuracy and number of deaths are given. In the last row a total performance is given for each of the situations.

4 Conclusion

From the classifier cross-validation scores in the first experiment we conclude that the piano task is the best task. The random task also performs very well. When we compare these on the game, we find that the random task has a higher accuracy and a higher score, as well as fewer deaths. Since classifier trained on the piano task was possibly not trained on the correct data, we will refrain from any conclusions here.

The classifier trained in the noisy environment already has a higher cross-validation score than the classifier trained in the normal environment. The performance on the game also turns out to be much better. The noisy environment lead to a higher accuracy and a higher score as well as fewer deaths. Again, since the classifier trained on the normal calibration was possibly not trained on the correct, data, we cannot draw any conclusions from this.

Comparing momentum encoding to position encoding, we see that, even though the momentum encoding is more physically plausible, it demands more precision from the classifier to work equally well as position encoding. We stopped this experiment early because of time constraints and the inability of the user to control the cannon demonstrated in the first two games. We do suspect that momentum can be useful for very experienced users.

5 Discussion

How valid are these conclusions? The corrupt data does not allow us to draw any conclusions, but even if we could have drawn conclusions, there are some important points to consider. Since we used only a single subject, there is nothing general that we can conclude from these data. In BCI research it is not uncommon to have only a few subjects for a study, but we agree that one subject is not enough.

Also within the constraints, this research could have been implemented differently. Our choices for the Buffer BCI environment, a TMS Mobita EEG cap with 10 electrodes and the space invaders game were inspired by constraints of the assignment. The implementation of the standard calibration environment was already available and therefore easy to compare with our own implementation. The preprocessing pipeline as well as the classifier implementation were inspired by the lectures on this subject, and assumed to be functioning well enough for our purposes.

In the first experiment we picked the best task based on classifier cross-validation score only. This again was because of time constraints. There simply was not enough time to compare the performance on the game for each task separately. Why we chose to randomize the tasks instead of pseudo-randomizing them needs some careful argumentation. Since we wanted to get rid of all predictability, as not to influence our subject's brain when calibrating, choosing pseudo-random would not have been random enough. Using five tasks and only forty trials, would have made the probability of a task being the next one too easy to predict; especially near the end of a sequence. Purely random tasks do not have this predictability. Using five sequences of forty trials also made the chances of having an unequal spread over the tasks smaller. Still, to eliminate the unevenness of the occurrence of the tasks, a bigger dataset needs to be created.

Slicing the single tasks from the random data should prevent learning or sequential effects. In the last session, we used the previously gathered data to train the classifiers and play the game. We did this, such that train-test transfer would be similar for all tasks.

We considered it unfair to play some of the games directly after calibrating and the rest of the games in another session. Since the calibration was quite lengthy, it was impossible to measure everything in a single session. Even in the game playing session we noticed that our subject became tired after playing the game for two hours without substantial breaks. This could have influenced the performance.However, because we did not perform the measurements sequentially per task, we believe the effects of tiring should be equal among all tasks. We thought it would be fair to first gather the calibration data.
Due to time and material constraints, it was impossible to gather more data. We realize that the data presented is not enough for statistically significant results. We do hope that the data is in fact the correct data and the comparisons made in this report are valid. However, since we cannot be sure about this, we draw no conclusions. More data needs to be gathered, preferably with more subjects. Consider this study as a pilot study, in which directions for research are identified and possibly discarded.

It is hard to say what went wrong with the data. The file that should contain the correct data, gave entirely different classifier scores and AUC plots. We suspect, based on the AUC plot, that the electrodes were connected in the wrong place. It is still not clear why the data is giving different results in the last session than in all sessions before. Unfortunately, due to time constraints, it was impossible to gather new data. Still, the research we did was not in vain. The results can clearly show that the type of motor task has some influence on the performance of the BCI.

References

Bartolic, E., Basso, M., Schefft, B., Glauser, T., and Titanic-Schefft, M. (1999). Effects of experimentally-induced emotional states on frontal lobe cognitive task performance. *Neuropsychologia*, 37(6):677–683.

Bauer, R. and Gharabaghi, A. (2015). Estimating cognitive load during self-regulation of brain activity and neurofeedback with therapeutic brain-computer interfaces. *Frontiers in behavioral neuroscience*, 9.

Beisteiner, R., Höllinger, P., Lindinger, G., Lang, W., and Berthoz, A. (1995). Mental representations of movements. brain potentials associated with imagination of hand movements. *Electroencephalogr Clin Neurophysiol.*, 96(2):183–193.

Cui, R., Huter, D., Lang, W., and Deecke, L. (1999). Neuroimage of voluntary movement: Topography of the bereitschaftspotential, a 64-channel dc current source density study. *NeuroImage*, 9(1):124–134.

Curran, E. A. and Stokes, M. J. (2003). 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.

Jayaram, V., Alamgir, M., Altun, Y., Scholkopf, B., and Grosse-Wentrup, M. (2016). Transfer learning in brain- computer interfaces. *IEEE Comp Intell Mag*, 11(1):20–31.

Lotte, B. (2015). to minimize or suppress calibration time in oscillatory activity-based brain computer interfaces. *proceedings of the IEEE*.

Miller, K. J., Schalk, G., Fetz, E. E., den Nijs, M., Ojemann, J. G., and Rao, R. P. N. (2010). Cortical activity during motor execution, motor imagery, and imagery-based online feedback. *Proc. Natl. Acad. Sci. USA.*, 107(9):4430–4435.

Wolpaw, J. R. and McFarland, D. J. (2004). 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.