ulm university universität
uulm

Faculty of Engineering, Computer Science and Psychology
Institute of Neural Information Processing

Neurotechnology: Brain-Machine-Interfacing Term Paper

Summer Term 2021

Presented by:
Carolin Schindler
carolin.schindler@uni-ulm.de

Primary instructor:
Prof. Dr. Dr. Daniel Alexander Braun

Secondary instructors:
Dr. Cecilia Lindig Leon
M. Sc. Sonja Schach

Last updated September 30, 2021

1 Brain-Computer-Interface (BCI)

“A BCI is a system that measures CNS [(central nervous system)] activity and converts it into artificial output that replaces, restores, enhances, supplements, or improves neural CNS output and thereby changes the ongoing interaction between the CNS and its external or internal environment.” [17, p. 3]
It is important to note that the activity of the central nervous system consisting of brain and spinal cord is recorded. Hence, BCIs are to be distinguished from neuroprostheses which instead use the activity of the peripheral nervous system including ganglia, nerve fibres and receptors in the body.

In this work, we have a more detailed look at three BCI experiments that utilize different types of signals recorded via electroencephalography (EEG). Before we dive into the main part, we give an overview over the general steps involved in converting the recorded CNS activity into an artificial output. Additionally, we outline the functional principles of EEG recordings.

1.1 Signal Processing Pipeline

A recorded brain signal is transformed into an artificial output for the respective application via three main steps [17] as visualised in Figure 1:

Signal Acquisition In this phase one measures the potential differences in the CNS activity using some signal recording technique with a reference voltage. Thereby, physiological and technical artefacts are recorded along with the actual signal leading to noise. In order to be able to process the signal digitally with a computer, we need to transform it from the analog domain (time and amplitude are continuous) to the digital domain (time and amplitude are discrete). For this purpose an analog-digital converter is suited [3], which is sampling under consideration of the

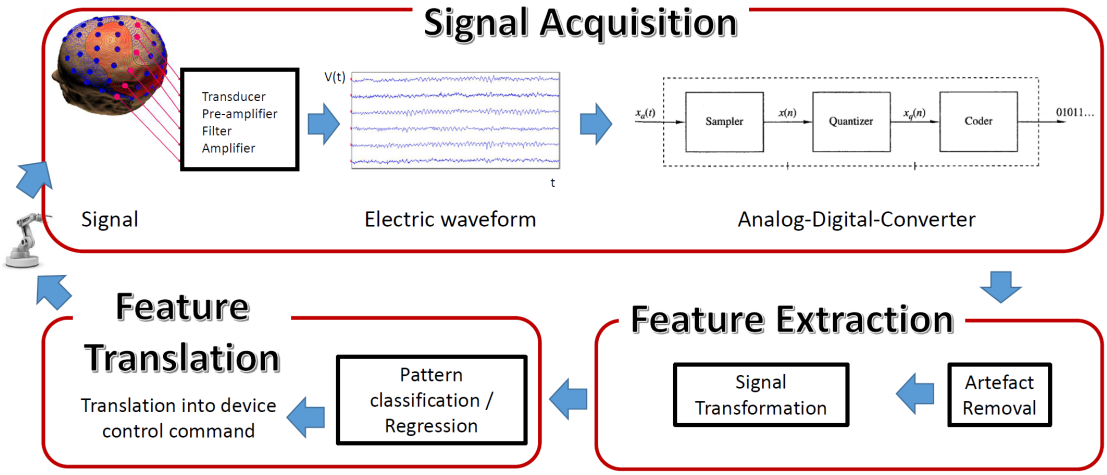

Figure 1: Main steps of the signal processing pipeline including a closed-loop control as the application gives feedback to the user.
[Source: lecture slides]

Nyquist-Shannon sampling theorem, quantizing and coding. The data we received for the experiments have this stage of processing.

Feature Extraction First, we have to think about removing the artefacts that were recorded in the signal acquisition phase. Possible methods include digital filtering (e. g. the line noise of 50 Hz can be removed with a notch filter) and spatial filtering (e. g. Laplace filter, principal component analysis, common spatial patterns). We are using digital filters in all three experiments and we apply the method of common spatial patterns in the Motor Imagery experiment (Chapter 4). For a full report on (EEG) artefact removal techniques, we refer to [15]. Further, the signal cannot only be observed in the time domain, it can be transformed into the frequency domain, which is the key to processing the data in the SSVEP experiment (Chapter 2).

Feature Translation The main challenge of this step is the training of a machine learning (ML) model [2] (e. g. Fisher discriminant analysis, linear regression) with the extracted features. In our case, all experiments are classification problems meaning that we have to assign a quantitative/categorical label to each recording. In order for the application to be controlled by the output of the ML model, we would have to translate the labels into respective control commands.

As the feature extraction and translation phases have some basic methods that are common for all experiments, I created respective Matlab toolboxes: The “ToolboxFeatureExtraction” contains methods to plot the recorded signals in the time-

and frequency-domain and calculation functions for the respective transformations. With this, we can perform a time-frequency analysis and extract custom features. Moreover, the common spatial patterns algorithm for automated feature extraction is implemented here. The “ToolboxClassification” is applied to perform cross-validations and to train, evaluate and apply the ML models stepwise linear regression and Fisher discriminant analysis.

1.2 EEG Recording

As mentioned above, the non-invasive EEG (electroencephalography) recording technique [1, 16, 12] was employed to measure the CNS activity. The subjects are wearing a cap with EEG electrodes. Hence, the electric signal generated by the population of neurons, the pyramidal cells to be more precisely, has to be transmitted through the biological tissue and the skull (= volume conduction) resulting in additional noise. An EEG can only record from the outer layers of the brain, thereby having a good temporal resolution in the millisecond range but a poor spatial resolution in the square centimetre range. The obtained recordings are especially useful for capturing oscillations in the brain activity in different frequency bands of which some are depicted in Figure 2.

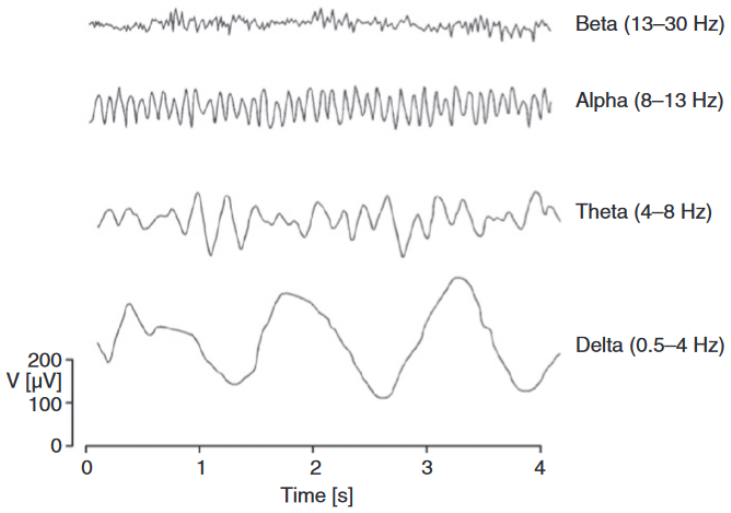

Figure 2: Different frequency bands in the EEG recorded signal.
[Source: [12] adapted from http://www.bem.fi/book/13/13.htm]

Different classes of EEG signals and hence potentials can be recorded. Among them are stimulus-evoked potentials and oscillatory potentials. The former requires an externally-paced BCI where the presentation of an external stimulus is necessary in order to trigger certain stereotypical EEG responses, such as the SSVEP (Chapter 2) or the P300 potential (Chapter 3). The oscillatory potentials can be triggered by people themselves through internal processes, hence the BCI is self-paced in this case. An example for these potentials is event-related desynchronization which is observable in Motor Imagery experiments (Chapter 4).

2 Steady-State Visual Evoked Potentials (SSVEP)

In this experiment, two buttons flickering with different but fixed frequency were presented to the subject in order to answer questions with 'yes' or 'no'. The implementation can be found in the "SVVEP.m" file.
The first evidence that the frequency of the flickering one is attending to can be captured in EEG recordings and thus can be exploited for BCI control was provided by [13]. Thereby, the repetition rate of the stimulus has to be larger than 5Hz and one should not select harmonics of frequencies for different targets as otherwise the targets are not distinguishable. The potentials that are visible in the EEG recording due to this stimulation are called Steady-State Visual Evoked Potentials (SSVEP). The target the user is attending to can then be found by transforming the EEG signal into the frequency domain and identify the frequency (and potential harmonics) with the highest amplitude in the power spectrum.

2.1 Analysis and Results

Usually, the repetition rates of the stimulation are known by the experimenter or the designer of the application. However, as we did not conduct the experiment by our own and only have the generated data, we first have do identify the two frequencies that were used for the 'yes'- and the 'no'-button. We can do this in general, by transforming the recorded signal into the frequency domain, using e. g. the fast Fourier transformation (FFT) [7], and search for peaks (that are not harmonics) in the power spectrum. The additional knowledge that the slower frequency is associated with the 'yes'-button, allows us to assign the identified repetition rates to the buttons and to label each trial with either 'yes' or 'no' in an unsupervised manner.

determine freq_slow and freq_fast We exclude the first second of each trial as we assume that this second mainly reflects activity due to stimulus onset. Further, we bandpass filter the signal into the frequency range from 5Hz to 45Hz since the stimulation rate for an SSVEP experiment should be above 5Hz and we also want to filter out the 50Hz line noise. Moreover, we average over the trials, which reduces the noise as the recorded signals are time-locked. Then we apply the FFT, which is a computationally efficient realisation of the discrete Fourier transform (DFT). The DFT transfroms a discrete time signal into a discrete frequency signal under the assumption that the signal in the time-domain is periodic: $X[n] = F_D\{x[k]\} = \sum_{k=0}^{N-1} x[k] \exp(-j2\pi \frac{kn}{N})$. As the periodicity assumption does not hold in our case, we need to apply a window function in order to reduce the leakage problem (= discontinuities and smearing effects). This frequency signal is transformed into the power spectrum, which representing the energy of the signal over the frequencies, by taking the absolute values (for energy representation one would additionally have to square). In order to reduce the noise further, we average over the 12 channels that were recorded for each trial. We now have single power spectrum for all the trials and all channels. Figure 3 shows the result of all these steps and we can even visually determine the searched frequencies: freq_slow is represented by the peak at 8Hz and we can also observe the harmonic frequency at 16Hz, freq_fast is at 14Hz.

label the trials With the above step, we have learned the characteristics of the data in an unsupervised manner. Now, we filter, apply the FFT, average over the channels and compute the power spectrum as above (we only leave out the step

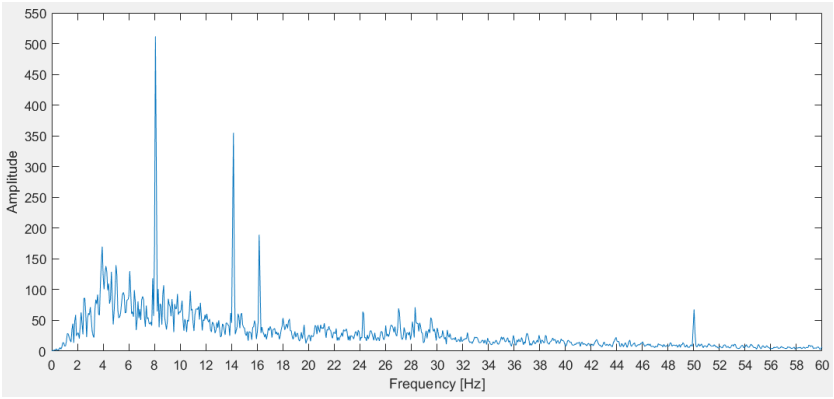

Figure 3: Power spectrum to determine freq_slow (=8Hz) and freq_fast(=14Hz).
[Source: generated with own Matlab Code]

where we average over trials). For this pre-processed signal, we identify for each trial the frequency with the maximum amplitude as a feature. The hand-crafted classifier simply determine whether the feature frequency is closer to freq_slow or freq_fast and assigns the respective label ‘yes’ or ‘no’ to the trial.
This leads to the following sequence of answers for the given data:
‘yes’, ‘no’, ‘no’, ‘no’, ‘yes’, ‘yes’, ‘yes’, ‘no’, ‘yes’, ‘yes’

2.2 Discussion

In Figure 3, we see why targets are not distinguishable when harmonic frequencies are chosen. We can also see that the bandpass filtering is not applicable perfectly in reality as we can e. g. still identify a smaller peak for the 50Hz noise.
An advantage of this procedure is that we do not require labelled training data and hence also do not need to train a classifier specific to a single individual (as it is the case for the P300 in Chapter 3 for example). When we know the repetition rates of the targets, we can place anyone in front of our BCI and evaluate their SSVEP to find out the target they focused at.
In our example we only had two targets, but in e. g. [4] the method was applied to 13 different buttons that were flashed at different frequencies building a number pad. Across all subjects they reached an average information transfer rate of 0.4524 bits/sec.

3 P300 Speller

The P300 Speller [6] is a spelling device that arranges the letters and numbers in a 6×6 grid – the Donchin matrix as shown in Figure 4. In order to spell a symbol, a row and a column has to be select. This is achieved, by highlighting each row and column e. g. 10 times while the user focuses on the symbol they want to write. Whenever the attended symbol is flashed up with the respective row or column, the user elicits a P300 potential also known as surprise or oddball potential: During the highlighting procedure only 20 out of 120 times, the selected letter will light up and hence this is an infrequent but relevant event. Thus, the task is to classify responses as frequent or infrequent and then determine the row and column of the two infrequent responses leading to the aimed letter.

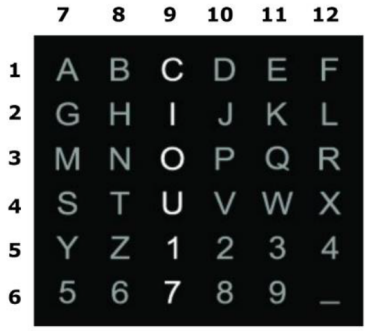

Figure 4: Donchin matrix representing the spellable symbols in a 6×6 grid.
[Source: Exam tasks]

The P300 potential is one of the event-related potentials (ERP) that can be observed when EEG responses that are time-locked to an event are averaged over many trials. Their importance for BCI interfaces arises due to the fact that they are typical even across subjects and hence can be used for decoding mental states. The P300 for example typically shows up ≈300ms after onset of an oddball stimulus. However, in order successfully create ERP-based BCIs, supervised training of the classifier for individual subjects is required as there is still some deviation.
The implementation of this experiment is in the “P300Speller.m” file.

3.1 Analysis and Results

In this experiment we not only try to predict the spelled words in the training and in the testing data, we also compare the performance of two different binary classifiers: The Fisher discriminant analysis (FDA) [5, 14] maximizes the Fisher criterion $J(\vec{\omega}) = \frac{\vec{\omega}^T S_B \vec{\omega}}{\vec{\omega}^T S_W \vec{\omega}}$ which is the ratio of the between-class difference and the within-class variance. Hence, this methods tries to learn a projection $\vec{\omega}$ such that the means of the two feature distributions are as far away as possible while taking the variances into account. In our case the FDA is applied with hand-crafted features. The linear regression model [2] assumes a linear relationship between the EEG response $\vec{x}$ and the classified label $y = \omega_0 + \sum_i^d \omega_i x_i$. We input the complete EEG responses as we apply a stepwise linear regression method which adds and removes variables depending on statistical significance evaluations.

Initially, we group the training data into the frequent and infrequent response classes and respectively reduce or zero-pad the trials to the median length of all trials, which is 64 samples corresponding to a recording time of approx. 252ms.

In order to extract the hand-crafted features for the FDA, we have a look at the event-related potentials of each class: We filter out the delta-band (ranging from 0.5Hz to 4Hz) and average over all the samples per class. The resulting signal for channel 6 and 8 is shown in the upper part of Figure 5. There we can see the P300 response for the infrequent trials (blue curve) already at around 200ms after the stimulus onset. The additional correlation plot helps us to identify the channels and time areas in which to extract the features. The feature extraction was implemented as a separate function in the "P300Speller_createFeature.m" file. For completeness, we also had a look at the signals in the frequency-domain and plotted the spectrogram.

In a 5-fold cross validation on the training data, we compare the performance of the FDA with the stepwise linear regression. The results are noted in Table 1. When taking the variance into account, the two models performed nearly the same in the cross-validation. However, we can see that the stepwise linear regression achieves an higher accuracy than the FDA when trained on the complete training data. This implies that the hand-crafted features extracted for the FDA may have room for improvement. We choose the linear regression model to try to predict the symbols spelled by the subject.

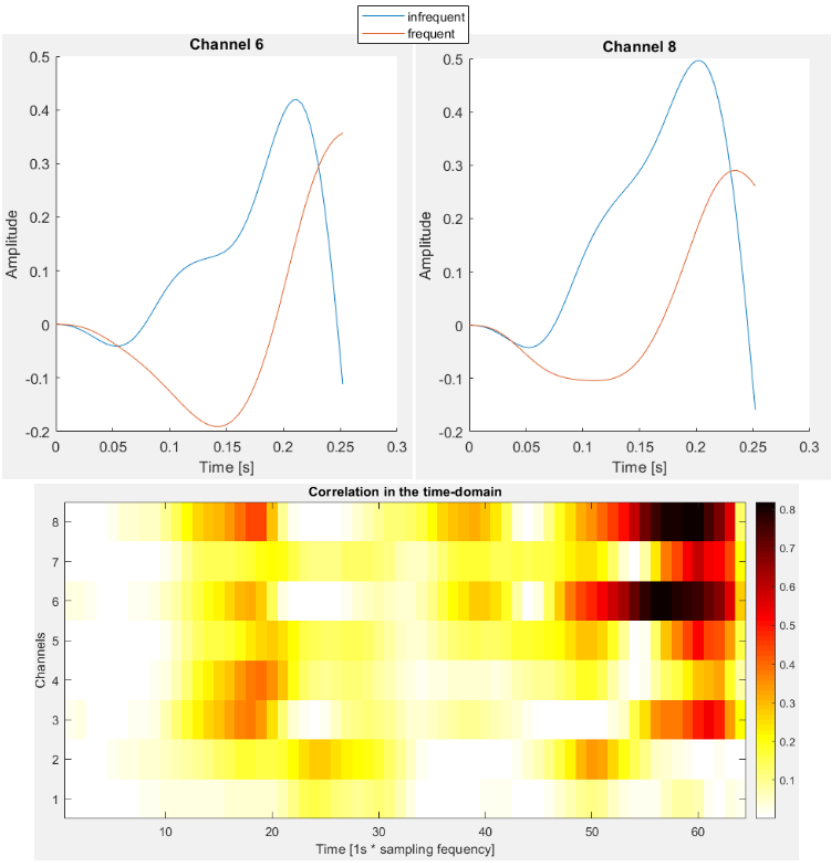

Figure 5: upper part: event-related potentials for channel 6 (electrode PO7) and channel 8 (electrode Oz)
lower part: correlation between ERP and class label for each channel
[Source: generated with own Matlab Code]

	mean accuracy	**variance**	**accuracy (complete data)**
Fisher discriminant analysis	0.86	0.028	0.86
stepwise linear regression	0.88	0.049	0.95

Table 1: Result for the two models in the 5-fold cross-validation and when training on the complete training data.

Table 2 shows the letters predicted by the stepwise linear regression model trained on the complete training data grouped into Spanish words.

training	**predicted word**	**actual word**	**English translation**
	TOKEN	TOKEN	token
	MIRAR	MIRAR	to look
	JUJUY	JUJUY	jujuy
	SANSO	MANSO	gentle
	CINCP	CINCO	five
testing	**predicted word**	**potential Spanish word**	**English translation**
	HUEGOQUE	JUEGO QUE	play something

Table 2: Spanish words spelled with the P300 Speller predicted with the stepwise linear regression model.
[translation into English via google translator: https://translate.google.com/?hl=de&sl=es&tl=en&op=translate]

3.2 Discussion

Usually, one would record the EEG signal for the training not only for 252ms in the median. In this case, it did not matter as we could observe the P300 potential at around 200ms. But as mentioned above, usually the P300 is elicited ≈300ms after stimulus onset and there might be also be persons showing this potential later than that. Hence, it is a good idea to record a longer period for the training as we can then cut the signal after the occurrence of the P300 and train our classifiers with this 'cutted' EEG response.
The P300 Speller in [6] was able to achieve an information transfer rate of up to 0.20 bits/sec.
In general, the P300 signal can also be exploited for BCIs that do not require the selection of a target, e. g. for lie detection [8].

4 Motor Imagery

In a motor imagery experiment [12, 9], the subject is instructed to give the command to execute a movement but to not actually executed it. This leads to activation in the brain areas as if one would actually move the imagined body parts. In our case the movement of either the left hand, the right hand, both hands or both feet was imagined. The motor homunculus shown in Figure 6 illustrates the disproportionate representation of certain body parts in the motor cortex and their topographic mapping (= their location within the motor cortex resembles their actual locations in the body). This spatial separation of the different body parts in the brain suggest to apply spatial filtering in order to automatically find out the electrodes that are relevant for the imagined movements of certain body parts. However, the body parts are more distributed than suggested by the homunculus.

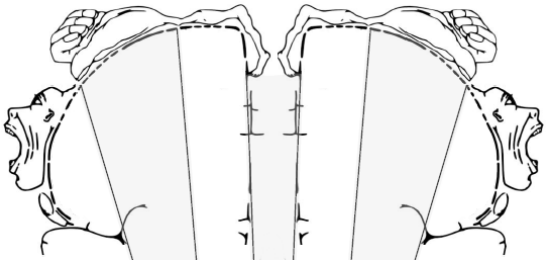

Figure 6: Motor homunculus. [Source: [9]]

4.1 Analysis and Results

In this analysis, we compare the performance of a Fisher discriminant analysis (FDA, see P300 Speller (Chapter 3) for brief explanation) with hand-crafted features and features created automatically with the common spatial patterns (CSP) algorithm. Further, we have a four-class classification problem that we want to solve with binary classifiers only. The implementation is done in the file “MotorImagery.m”. In a preprocessing step we filter the signals to the alpha- and beta-band (8Hz to 30Hz) and remove the first second of each trial.

four-class classification problem There exist different approaches to solving n-class problems with binary classifiers [2]. One vs. all. trains n binary classifiers where each classifier separates one class from all the other n-1 classes. This leads to training binary classifiers with unbalanced data which commonly impairs performance. Another idea is to use an hierarchical approach: In our case one could imagine to firstly decide whether two body parts (both hands or both feet) or a single body part (right hand or left had) were imagined to be moved. Afterwards this stage we use a respective binary classifier to clarify on the remaining two classes. The training data for each binary classifier would be balanced, however there is a large dependence on the first classifier to be correct as we otherwise input the data into the wrong follow-up classifier. The approach we applied hence, is one vs. one (OvO): Hereby, we train $(n-1)\cdot(n-2)$ different binary classifiers that distinguish between each possible pair of classes. The final selection of the class label is the performed via majority vote and one has to think about conflict resolution in case there are several labels sharing the majority. In our specific experiment we have four different classes leading to six binary classifiers. Possible conflicts are among three or two labels: When the conflict is among three labels we choose one label randomly (this could potentially be improved), when the conflict is between two labels we query the respective binary classifier and take the label predicted by this. The OvO model is implemented in the files "MotorImagery_train_OVO.m", "MotorImagery_predict_OVO.m", "MotorImagery_eval_OVO.m".

hand-crafted features This time, the signals are not time-locked. Hence, we have to apply the transformations first and then average afterwards. In the time-domain we calculate the event-related desynchronisation (ERD) [10]: We choose a window with a certain size that we move along the time signal with a certain step size and in each window we calculate the squared mean of all trials within that window. For completeness, we also transform the original time signals into the frequency domain, however the correlation of the ERD with the class labels for each binary classifier is much higher than for the signal in the frequency domain. Analogous to the P300 Speller experiment (Chatper 3) we choose the channels and time windows of the ERP according to the plotted correlation.

CSP features Common spatial patterns (CSP) [11] is a data-dependent spatial filtering method. The idea is to reduce the variance in one class, while increasing it in the other. Hence features can be selected according to the variance of the data.

For the implementation, I refer to the “ToolboxFeatureExtraction”.

training and prediction Table 3 lists the results for training in a 5-fold cross-validation and for training with the complete training data. The left part of the table shows the accuracy values for the hand-crafted features and the right part for CSP features with three filter pairs (one and two filter pairs performed worse). We can read the values for single binary classifiers and for the classifiers applied in the one vs. one model for multi-class classification. The FDA with the hand-crafted values

	hand-crafted features			CSP features		
	mean accuracy	variance	accuracy (complete data)	mean accuracy	variance	accuracy (complete data)
bothHandVSbothFeet	0.95	0.067	0.95	0.88	0.085	1
bothHandVSleftHand	0.92	0.000	0.92	0.83	0.139	1
bothHandVSrightHand	0.83	0.053	0.85	0.68	0.170	0.97
bothFeetVSleftHand	0.88	0.041	0.88	0.70	0.113	0.98
bothFeetVSrightHand	0.95	0.067	0.95	0.87	0.041	1
leftHandVSrightHand	0.92	0.000	0.92	0.90	0.097	1
one vs. one model	0.78	0.041	0.79	0.58	0.112	0.96

Table 3: Result for the two different features in the 5-fold cross-validation and when training on the complete training data. The common spatial pattern features were extracted with three filter pairs.

performs better in the cross-validation, while the FDA with CSP features performs better when trained on the complete training data. This leads to the conclusion, that the CSP features represent the training data very well but cannot generalize to unseen data. Hence, we apply the FDA with hand-crafted features to predict the labels of the yet unlabelled testing data and obtain the following sequence of labels:
“both hands”, “both hands”, “both hands”,“both hands”,
“both hands”,“both hands”, “right hand”, “both hands”,
“left hand”, “right hand”, “both hands”, “left hand”,
“both hands”, “both feet”, “both hands”, “right hand”,
“left hand”, “both hands”, “left hand”, “both feet”,
“both feet”, “right hand”, “both feet”, “both hands”,
“both hands”, “both hands”, “both hands”, “both feet”,
“right hand”, “both feet”, “both hands”, “both hands”,
“left hand”, “right hand”, “both feet”,“both hands”,
“both feet”, “both hands”, “both feet”, “both hands”

4.2 Discussion

This kind of BCI requires training by the human as the motor commands need to be only imagined but not executed. Further, the BCI might need to be readjusted/retrained over time as the motor cortex (as any other brain areas) is affected by plasticity. This means that the electrodes start to capture different signals than compared to the training phase and hence the extracted features potentially are not working with the earlier trained classier.

Bibliography

[1] Hans Berger. “Über das Elektrenkephalogramm des Menschen”. In: *Archiv für Psychiatrie und Nervenkrankheiten* 87.1 (Dec. 1929), pp. 527–570. DOI: 10.1007/bf01797193.

[2] Christopher M. Bishop, ed. *Pattern recognition and machine learning*. Information science and statistics. Springer, 2006. ISBN: 9780387310732.

[3] Martin Bossert. *Einführung in die Nachrichtentechnik*. Oldenbourg, 2012. ISBN: 978-3-4867-0880-6.

[4] Ming Cheng, Xiaorong Gao, Shangkai Gao, and Dingfeng Xu. “Design and implementation of a brain-computer interface with high transfer rates”. In: *IEEE Transactions on Biomedical Engineering* 49.10 (2002), pp. 1181–1186. DOI: 10.1109/TBME.2002.803536.

[5] R. A. FISHER. “THE USE OF MULTIPLE MEASUREMENTS IN TAXONOMIC PROBLEMS”. In: *Annals of Eugenics* 7.2 (Sept. 1936), pp. 179–188. DOI: 10.1111/j.1469-1809.1936.tb02137.x.

[6] L.A. Farwell and E. Donchin. “Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials”. In: *Electroencephalography and Clinical Neurophysiology* 70.6 (1988), pp. 510–523. DOI: https://doi.org/10.1016/0013-4694(88)90149-6.

[7] Thomas Frey and Martin Bossert. *Signal- und Systemtheorie*. 2nd ed. Vieweg + Teubner, 2008. ISBN: 978-3-8351-0249-1.

[8] Syed Haider, Malik Daud, Aimin Jiang, and Zubair Khan. “Evaluation of P300 based Lie Detection Algorithm”. In: *Electrical and Electronic Engineering* 7.3 (July 2017), pp. 69–76.

[9] Cecilia Lindig León. “Multilabel classification of EEG-based combined motor imageries implemented for the 3D control of a robotic arm”. PhD thesis. Université de Lorraine, 2017.

[10] G. Pfurtscheller and F.H. Lopes da Silva. “Event-related EEG/MEG synchronization and desynchronization: basic principles”. In: *Clinical Neurophysiology* 110.11 (1999), pp. 1842–1857. DOI: https://doi.org/10.1016/S1388-2457(99)00141-8.

[11] H. Ramoser, J. Muller-Gerking, and G. Pfurtscheller. “Optimal spatial filtering of single trial EEG during imagined hand movement”. In: *IEEE Transactions on Rehabilitation Engineering* 8.4 (2000), pp. 441–446. DOI: 10.1109/86.895946.

[12] Rajesh P. N. Rao. *Brain-Computer Interfacing: An Introduction*. Cambridge University Press, 2013. ISBN: 978-0-521-76941-9.

[13] T.A. Skidmore and H.W. Hill. “The Evoked Potential Human-computer Interface”. In: *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. Vol. 13. 1991.

[14] Alaa Tharwat, Tarek Gaber, Abdelhameed Ibrahim, and Aboul Ella Hassanien. “Linear discriminant analysis: A detailed tutorial”. In: *AI Communications* 30.2 (May 2017), 169–190. DOI: 10.3233/AIC-170729.

[15] Jose Antonio Urigüen and Begoña Garcia-Zapirain. “EEG artifact removal—state-of-the-art and guidelines”. In: *Journal of Neural Engineering* 12.3 (Apr. 2015). DOI: 10.1088/1741-2560/12/3/031001.

[16] J J Vidal. “Toward Direct Brain-Computer Communication”. In: *Annual Review of Biophysics and Bioengineering* 2.1 (1973), pp. 157–180. DOI: 10.1146/annurev.bb.02.060173.001105.

[17] Jonathan Wolpaw and Elizabeth Winter Wolpaw, eds. *Brain-Computer Interfaces: Principles and Practice*. Oxford University Press, 2012. ISBN: 978-0-195-38885-5.