Real-time robot arm control using motor imaginary movements decoded from EEG signals

RESEARCH PRACTICE

submitted by
Juri Fedjaev

NEUROSCIENTIFIC SYSTEM THEORY
Technische Universität München

Prof. Dr Jörg Conradt

Supervisor:	Dipl.-Ing. Zied Tayed
Start:	24.02.2017
Final Submission:	22.05.2017

Abstract

People with severe neuromuscular disorders, such as late-stage amyotrophic sclerosis (ALS) and those paralyzed from higher level spinal cord injury are unable to actuate any of their muscles. Communication with the outside world is therefore problematic for the suffering people. Cognitive and sensory body functions, however, are often only minimally affected. Therefore, an electroencephalogram-based communication which does not require any neuromuscular control is considered to be particularly helpful to enhance the disabled's quality of life by increasing their independence.
A simple 3-channel brain-computer-interface (BCI) using motor imagery was developed for controlling a robot arm. Two classes representing left and right arm movements have been trained during recording sessions consisting of 40 and 50 trials. The participants were two healthy male students in the age of 24 and 25 years. Three bipolar electrode pairs were placed at C3, Cz and C4 locations for the recording of the EEG signals using a Biopac MP36 acquisition unit. A Matlab software was developed to interface with the Biopac hardware API, to generate visual cues for the BCI users and to record, process and classify the signals. Off-line classification accuracies of 77% were reached. On-line classification, however, was unsuccessful due to hardware limitations, especially the electrode design of the signal acquisition device.

Contents

Chapter 1

Introduction

People with severe neuromuscular disorders, such as late-stage amyotrophic sclerosis (ALS) and those paralyzed from higher level spinal cord injury are unable to actuate any of their muscles. Communication with the outside world is therefore problematic for the suffering people. Cognitive and sensory body functions, however, are often only minimally affected. Therefore, an electroencephalogram (EEG)-based communication which does not require any neuromuscular control is considered to be particularly helpful to enhance the disabled's quality of life by increasing their independence [1].

Besides EEG, there are other techniques for monitoring brain activity, such as functional magnetic resonance imaging (fMRI), magnetoencephalography (MEG), positron emission tomography (PET) or single photon emission computer computer tomography (SPECT). The advantages of those methods are a better accuracy and better spatial resolution compared to EEG. However, due to their large size, heavy weight and high price, they are not as suitable for BCI applications as EEG. Furthermore, EEG offers a better temporal resolution, portability and relatively low cost. Recently, low-cost consumer devices came on the market (e.g. EMOTIV EPOC+), which will further push the advancements in the area of EEG-based BCI [2].

In general, the information flow in a BCI follows the following path: signal acquisition, signal (pre)-processing, feature selection and extraction, classification (detection of distinct signal patterns), application interface (e.g. to a robot arm), and feedback (see figure 1.1).
For the project of this research practice, the main objective is to implement algorithms similar to those described by Meng and Yong [3, 4] to discriminate and decode four motor imagery movements (left hand, right hand, both hand imaginary movement and rest) from EEG signals. Afterwards, the classification has to be used to control a robot arm in a real-time scenario. Motor imagery (MI) is defined as the mental rehearsal or imagination of a physical movement without actually performing it physically [5]. On a neurophysiological level, similar brain regions are

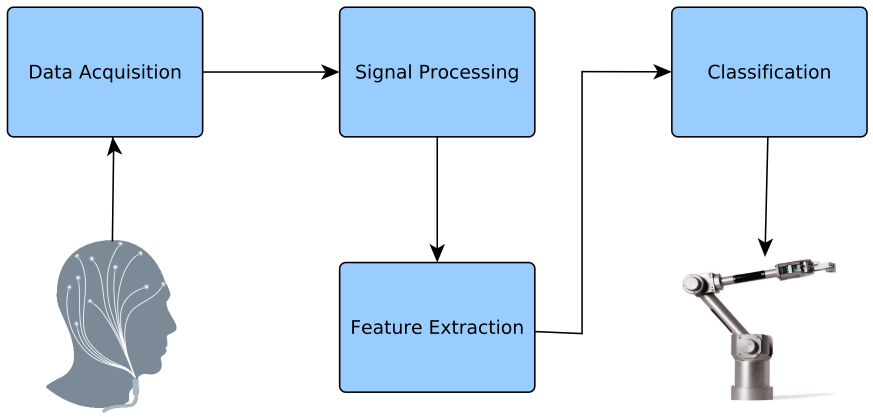

Figure 1.1: Information flow for a BCI system controlling a robotic arm.

activated during motor execution and motor imagery, however, the performance is blocked at a corticospinal level. Studies based on fMRI showed similar activation patterns during motor imagery and actual movement execution [6]. For operating a BCI, motor imagery has proven its capability as an efficient mental strategy.

Concerning the positioning of the EEG electrodes on the scalp, an internationally recognized method called the 10-20 EEG system is used (see fig. 1.2 for illustration). It was developed to ensure reproducibility by standardizing the electrode positions so that one subject's studies could be compared to each other. The system is based on the relationship between the location of an electrode and the underlying area of cerebral cortex. The "10" and "20" refer to the fact that the actual distances between adjacent electrodes are either 10% or 20% of the total front-back or right-left distance of the skull [7]. Previous studies show that electrode positions C3, Cz and C4 are suitable for recording characteristic motor imagery signals, as they are directly covering part of the sensorimotor area.

In the following, first the design and implementation of the BCI project will be discussed. Following that, the accuracies that have been reached will be presented and drawbacks of the approach and possible improvements will be discussed.

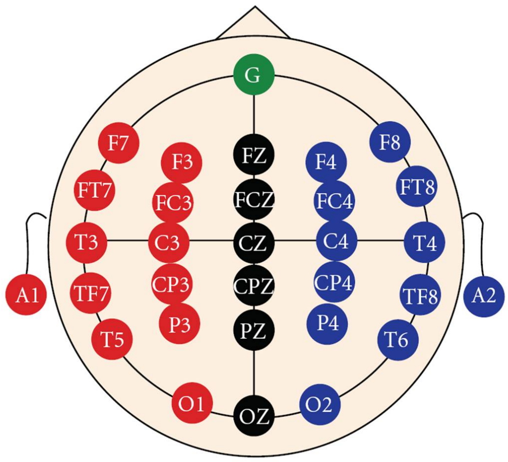

Figure 1.2: International standard 10-20 electrode placement system.

Chapter 2

Experimental Design and Implementation

The basic principle on which this BCI project for controlling a robot arm is based on is the following idea: First, a training dataset is recorded for subject that wants to execute control commands. Second, that data is used for training a classifier. Subsequently, the trained classifier model is saved and reused for on-line processing the subject's EEG data, enabling real-time robot control. In order to accelerate that procedure and minimize time constraints, there exist several frameworks for BCI applications. To name some of the most commonly used:

- BCILAB
  ◦ "Open-source MATLAB-based toolbox built to address the need for BCI methods development and testing by providing an organized collection of over 100 pre-implemented methods and method variants, an easily extensible framework for the rapid prototyping of new methods, and a highly automated framework for systematic testing and evaluation of new implementations" [8].

- OpenVibe
  ◦ "OpenViBE is a software for real-time neurosciences (that is, for real-time processing of brain signals). It can be used to acquire, filter, process, classify and visualize brain signals in real time. OpenViBE is free and open source software. It works on Windows and Linux operating systems" [9].

- BCI2000
  ◦ "BCI2000 is a software suite for brain-computer interface research. It is commonly used for data acquisition, stimulus presentation, and brain monitoring applications. BCI2000 supports a variety of data acquisition systems, brain signals, and study/feedback paradigms. [...] BCI2000 is available free of charge for research and education purposes" [10].

2.1 First approach: OpenVibe and Emotiv EPOC+

As previously mentioned, using an existing software framework accelerates development enormously. Therefore, the first approach was to use OpenVibe to implement the BCI system. This was possible because the EEG system for signal acquisition that was available in the beginning of the project was the Emotiv EPOC+, which offers driver support for its use with OpenVibe. The EPOC+ is a wireless, 14 channel-device with a sampling rate of 128 Hz (see fig. 2.1 on the right-hand side). Its electrodes cover many of the electrode positions as proposed by the 10-20 system (see above), thus also the area near the sensorimotor cortex at C3 / C4 locations.
OpenVibe's programming paradigm is based on building blocks representing individual signal processing algorithms which can be easily connected to each other. Using Python and/or C++, it is also possible to build new blocks with user specific algorithms (see fig. 2.1 on the left-hand side for an illustration). The interface to

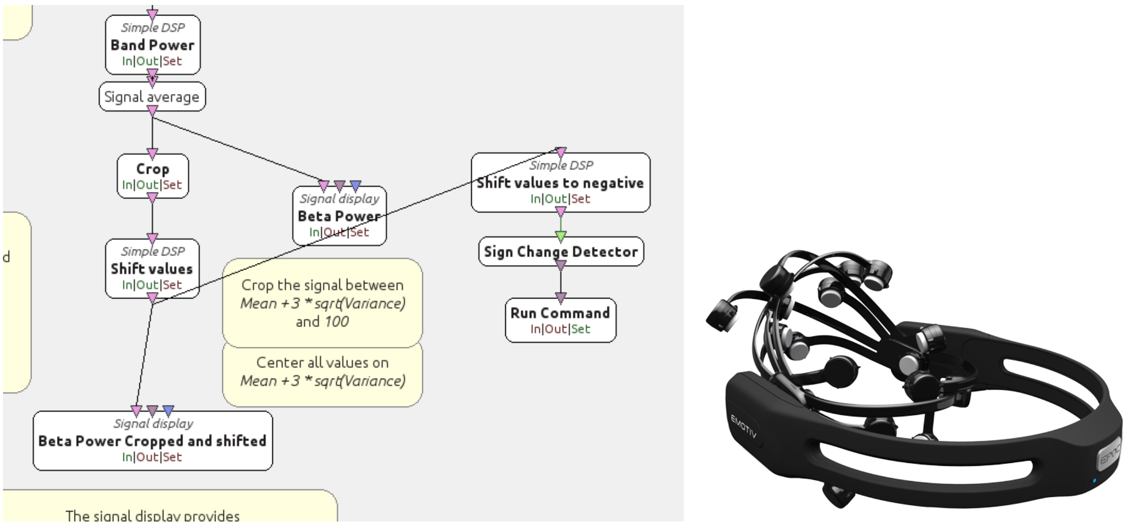

Figure 2.1: Left: Example for the work flow in OpenVibe using algorithmic building block. Right: The Emotiv EPOC+ headset for signal acquisition.

OpenVibe is the so-called *OpenVibe Acquisition Server*. For accessing the raw signals of the EPOC+, a premium software development kit (SDK) is needed. However, there are issues trying to connect the EPOC+ with the OpenVibe aquisition server. In fact, the connection succeeds but stops working when starting the acquisition (see fig. 2.2).
Two similar headsets with the official premium SDK were tested without success. Emokit reverse-engineering drivers written in Python (found on GitHub) were able to access some of the data. However, the electrode and quality indicators were disturbed by a strong jitter in the data. An Emotiv EPOC+ headset with an older version number was working without any problems, but was unfortunately available only briefly for testing purposes and not for the BCI project. Therefore, it is assumed

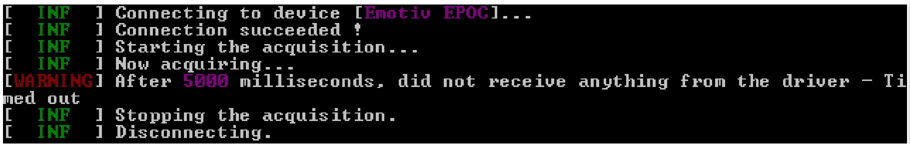

Figure 2.2: Error encountered when trying to connect the Emotiv EPOC+ with OpenVibe Acquisition Server.

that there were some changes done on firmware level by the manufacturer. In the end, the approach using an Emotiv EPOC+ in connection with OpenVibe was abandoned due to the previously described problems. The alternative solution will be explained in the following section.

2.2 Second Approach - Biopac MP36

Following the difficulties using the Emotiv headset, alternative hardware for EEG signal acquisition was needed. Fortunately, NST has two Biopac MP36 systems available. The Biopac MP36 is a biosignal acquisition device specifically developed for undergraduate students studying physiology. It is distributed as a complete solution called *Biopac Student Lab* (BSL). The BSL system integrates hardware, software and curriculum materials including experiments that students use to study the cardiovascular system, muscles, pulmonary function, autonomic nervous system, and the brain [11].
The MP36 has four analog input channels, each with separate ground and reference electrodes. It has an A/D sampling resolution of 24-Bit and a maximum sampling rate of 100 kHz per channel, see figure 2.3.

Figure 2.3: Left: Biopac MP36 data acquisition unit. Middle: Biopac electrode connection cable with separate reference and ground per channel. Right: Disposable Biopac electrodes.

As the Biopac system is supposed to be used with the proprietary Biopac Student Lab software, there is no BCI software available that supports the MP36 acquisition device natively. However, Biopac provides a hardware API (BHAPI) for developers. The BHAPI enables the user to acquire raw data, set sampling rate, triggers, get the status of the MP36 device as well as some other features. The implementation of these functions is compiled into a Windows 32-Bit DLL (dynamic link library) called `mpdev.dll` (see fig. 2.4). The interface is documented in C/C++, but any programming language that is able to utilize Windows 32-bit DLLs should be able to access the BIOPAC Hardware API. Due to personal preferences Matlab is used.

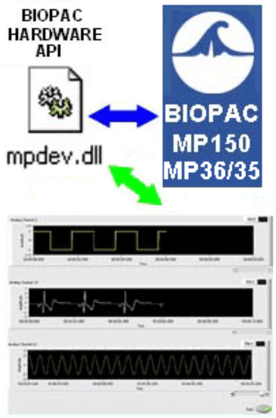

Figure 2.4: Illustration of the Biopac hardware API as an interface to external programming languages or software (here: LabView).

2.2.1 Data Acquisition

In order to train the classifier model, training data is necessary. Therefore, a cue-based experimental paradigm was used. The participants of the data recording sessions were shown visual cues in form of left or right arrows at distinct points in time for the onset of left or right arm motor imagery of movements (see figure 2.5), representing the two classes. One recording sessions consisted of 40 or 50 trials, 20 or 25 trials, respectively, for each side (left, right). One trial has a total duration of 8 seconds, with cue onset on second 3 for a duration of 2 seconds. During cue display, the participant is supposed to maintain the motor imagery accordingly (see figure 2.5 for illustration).
For the training sessions, a Matlab function `recordMotorImageryData(numTrials, nCh, cueOn)` has been written. It takes as input arguments the desired number of trials (`numTrials`), the number of channels used for recording (`nCh`), and whether a cue shall be displayed or not (`cueOn`). If a cue is not desired, the function will plot a graph of the EEG.

Launching data acquisition: In order to start the MP36 acquisition device, an interface to Matlab was written, namely the function `startAcquisition(dothdir, libname,mptype, mpmethod, sn, duration)`. It takes as inputs the path to the C/C++ header file `mpdev.h` (`dothdir`), the name of the Biopac API DLL file (`libname`), the type of Biopac device (`mptype`, has value '`103`' for MP36), the connection method `mpmethod` (has value '`10` for USB connection), the serial number

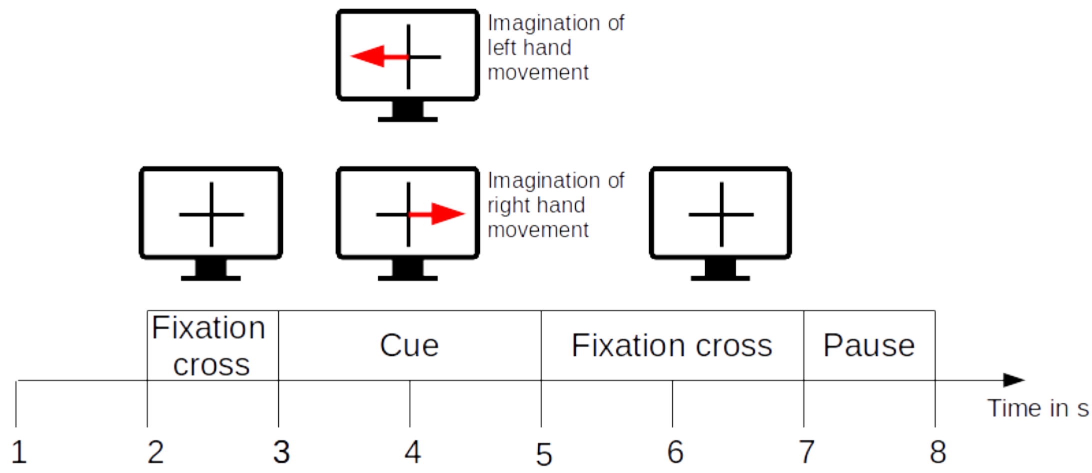

Figure 2.5: Timing of the movement imagery task. Visual cue stimulus in form of left/right arrows between 3 and 5 seconds instructs the participant o imagine the desired movement (of left/right arm, respectively).

of the Biopac device `sn` (can be put to '`auto`', if only a single Biopac device is connected to the computer), and finally the duration of data acquisition in seconds `duration`.

Cue display: For cue display, another Matlab function `launchCueExp(DURATION, T_BLANK, T_CUE_ON, T_CUE, T_PERIOD)` has been written. By modifying its input parameters, individual visual cue timings may be re-used for future experiments at NST. The inputs are the total experiment duration in seconds `DURATION`, the fixation cross onset `T_BLANK`, the cue onset time `T_CUE_ON`, the duration of cue display `T_CUE`, and the period length for one trial `T_PERIOD`. For illustration of the parameters see figure 2.5.

Data Format:

Electrode positioning

The electrode positioning on the scalp is a crucial part of BCI design. As mentioned before, electrode positions on C3, Cz and C4 are desired for motor imagery classification tasks. Therefore, each of the bipolar electrode pairs are positioned accordingly on those positions (see figure 2.6 in the middle for illustration). The ground electrodes may be positioned on the mastoid parts of the temporal bone, right behind the ears (see figure 2.6 on the left for illustration). As the electrodes used with the Biopac device are dry electrodes that are more suitable for EMG recordings, great caution must be exercised while placing them. Participants with thick hair may be a problem due to low signal-to-noise ratio (SNR) and high impedance. The subject's hair should be gently moved to the sides, so that the electrodes touch the scalp. After placement, the the electrodes should be pressed onto the scalp with minor force for roughly one minute each. Then, bandages were wrapped around the subjects head to fixate the electrodes. The impedance for the electrodes should ideally be lower than 10 kΩ, which can be checked using the "*Impedance Check*" port on the Biopac device. A value under 50 kΩ, however, was accepted during recordings, as it was hard to achieve a lower values.

As a side note, the recording setup with the Biopac device is very uncomfortable

Figure 2.6: Left: Electrodes on subjects scalp as seen from the left-hand side with ground electrode on the mastoid part of the temporal bone. Middle: View from the top showing C3, Cz and C4 electrode positions (l.t.r.) using bipolar derivation. Right: Subject's view and position for the recording session.

for the participant and was unsustainable for longer periods of time ($>$ 15 min). After the recording, the removal of the electrodes is tedious and painful because the adhesive that sticks strongly to the hair.

2.2.2 Signal pre-processing

Raw EEG signal is very noisy due to various factors, such as low SNR, artifacts from muscle movements (eye movements, cardiac signals, muscles beneath the scalp etc.). Therefore, the signal first needs to be prepared for further analysis. In many of the recordings, a baseline drift is noticeable across most of the channels due to varying signal quality (pressure from bandage not constant, electrodes slightly moving on the scalp, impedance not constantly the same for individual electrodes), see figure 2.7 on the top left corner for an example. Therefore, a function `detrendData(input, pos)` has been written. This function takes as input a data matrix `input` with a column for each channel and the samples along its rows. The input `pos` tells the function where the onsets of the trials are, so that trial-wise mean subtraction can

be applied.
Furthermore, the interesting frequency bands for motor imagery decoding are mostly in alpha- (8-13 Hz) and beta-bands (16-25 Hz) [12]. Bandpass filtering from 8 to 30 Hz is therefore applied (as distinct frequency subbands will be extracted in the feature extraction step, no further bandpass filtering is employed at this point). The filter used is a 8th order Butterworth filter from the Matlab toolboxes. Additionally, the EEG data is normalized and scaled. See figure 2.7 for the raw signals (left-hand side) and the corresponding filtered, detrended and normalized signals (on the right-hand side).

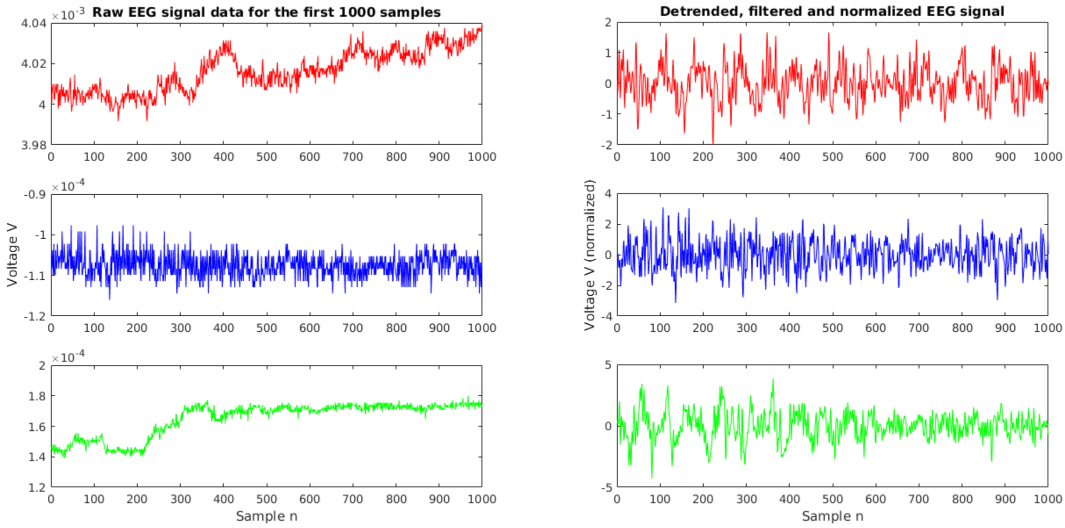

Figure 2.7: Left: First 1000 samples (5 seconds) of the raw signals from channels 1, 2 and 3. Right: Removed baseline drift, bandpass filtered and normalized signals from channels 1, 2 and 3.

2.2.3 Feature extraction

For feature extraction, epochs of 2000 ms length plus 500 ms pre-stimulus baseline were extracted from the EEG data in reference to the cue onset time. Afterwards, band power features (BP) were computed. Each of the three EEG channels was used to calculate BP features in 72 frequency bands using numerous overlapping narrow bands between 8 and 30 Hz, yielding a total of 216 BP features [13, 14]. As the dimensionality of the extracted EEG features is relatively high and the later used SVM classifier expects low-dimensional feature vectors, it is necessary to apply a dimensionality-reduction algorithm before feeding them in the classifier. Therefore, a conventional principal component analysis (PCA) algorithm, as it is implemented directly in Matlab is used. 12 PCA coefficients were used, leading to a 12-dimensional feature vector per trial. The feature extraction algorithms were adapted from the work of Tayeb and Ercelik [14].

2.2.4 Classification

The main goal of the BCI project is to translate recorded brain activity into a robot command. The last step in the control chain involves the identification of feature patterns in order to classify the user's intent. The output of the classification stage is the control input of the robot arm.
There are various classification algorithms that may be used to design BCI systems: linear classifiers, such as Linear Discriminant Analysis (LDA), Support Vector Machine (SVM), Artificial Neural Networks (ANN) and non-linear Bayesian classifiers. In general, LDA and ANN are commonly used for BCI tasks. One of the drawbacks of linear classifiers like LDA and SVM are that they may perform poorly for multi-class classification on the non-linear EEG data [2]. However, considering only two classes shall be discriminated at a first step, a SVM classifier will be used for the BCI system.
The Matlab toolbox for *Statistics and Machine Learning* provides a function `fitcsvm(_)` that takes a feature matrix and labels as inputs and trains a support vector machine model. There are various optional parameters that can be set, in this case a radial basis (RBF) kernel option was used to let the software find a scale value for the kernel function. Additionally, the predictors were standardized, which is good practice. The output is a SVM model that in a next was cross-validated using 10-fold cross-validation with the `crossval(SVMModel)` function. The cross-validated SVM model was then used to compute the misclassification rate with the function `kfoldLoss(CVSVMModel)`. For later use in an on-line scenario the SVM model is saved.

2.2.5 Robot Arm Control

The last stage in the control chain is the robot arm control. At NST, two Neuronics Katana arms are available. The Neuronics Katana has 6 degrees of freedom (including the gripper) and a payload of 500g, which is more than sufficient for the purposes of this project (see figure 2.8). Thanks to working student Bernhard Specht, who wrote a Python script for accessing the robot methods via SOAP server, the robot can be easily controlled. In order to control the robot arm via Matlab, a simple Python wrapper was built. In short, it may be used via `py.KatanaSoap.KatanaSoap()` to get a `katana`-object. With this object, several commands may be executed, such as

- `katana.calibrate()`
  - Calibrate the arm

Figure 2.8: 6-DoF robot arm Neuronics Katana.

- `katana.moveMotAndWait(axis, val)`
  ○ Move axis "`axis`" to position `val`. the `int32(val)`-function needs to be used in Matlab to ensure the functioning (the method only accepts integers; Matlab uses floats with double precision by default).

- `katana.closeGripper()` / `katana.openGripper()`
  ○ Closes or closes the gripper, respectively.

Further documentation can be found in the Python script `KatanaSoap.py` written by Bernhard Specht.

For the on-line application of the BCI system, a Matlab function `realtimeMotorImagery(numTrials, robotOn)` was written. It takes as inputs the number of trials `numTrials` to be attempted and whether the robot shall be actuated (input `robotOn`). Due to time constraints that resulted from the failed attempt using the Emotiv device in the first weeks of the research practice, a real-time application could not be realized. Instead, randomized cues (left arrows, right arrows) are displayed, EEG signal epochs extracted and directly classified, resulting in a control signal for the robot. In the following section the results will be presented and discussed.

2.3 Experimental Results

For training data, two-class EEG motor imagery data has been collected from two healthy male subjects in the age of 24 and 25 years. A total of 11 datasets has been recorded. Table 2.1 summarizes the cross-validated off-line classification accuracies reached with the SVM using BP features. Regarding the experimental design, there are several limitations. For example, the visual cue stimulation could be randomized, as in its current implementation left/right cues are alternating and therefore predictable for the user. Additionally, a pause of random length could be added at the end of each trials in order to avoid the user getting tired or bored during the recording session.

Recording ID	Number of trials	Overall Accuracy (%)
AS01-11	40	64
AS02-11	40	67
JF02-10	40	56
JF03-10	40	63
JF04-10	40	74
AS02-12	40	**77**
JF02-15	50	66
JF02-16	50	60
JF01-15	50	58
JF01-16	50	60
AS01-16	50	46

Table 2.1: Summary of cross-validated off-line classification accuracies reached with a SVM using BP features.

It is apparent that there is a high variability in the classification accuracy across the recording sessions. This is due to various reasons that will be discussed in the section following.

One of the results certainly is that the Biopac MP36 device is unsuitable for BCI applications, even though it delivers decent results for simple EEG recordings. To summarize the lessons learned while using the MP36:

- Electrode positioning has a big impact on BCI performance (as mentioned before). However, as Biopac does not include a EEG cap for well-defined electrode positions (electrodes need to be placed "individually" on the scalp; see figure 2.6), it is particularly difficult to reproduce results from session to session.
- Longer hair increases the impedance and therefore lowers the SNR. Additionally, it is harder to fixate the electrodes precisely on the scalp when using conventional elastic bandages.
- The adhesive part of the electrode should be softly abraded with abrasive pads in order to reduce the discomfort for the user of the BCI.

Despite the difficulties, there are two recorded datasets that deliver satisfying results, namely those with recording IDs `AS02-12` and `JF04-10` with 77% and 74% classification accuracy, respectively.

Chapter 3

Conclusion

A simple 3-channel BCI using motor imagery was developed for controlling a robot arm. Two classes representing left and right arm movements have been trained during recording sessions consisting of 40 and 50 trials. The participants were two healthy male students in the age of 24 and 25 years. Three bipolar electrode pairs were placed at C3, Cz and C4 locations for the recording of the EEG signals using a Biopac MP36 acquisition unit. A Matlab software was developed to interface with the Biopac hardware API, to generate visual cues for the BCI users and to record, process and classify the signals. Off-line classification accuracies of 77% were reached, even though in general the classification accuracy was lower around 60%, partly due to hardware limitations, especially electrode design.

Besides the hardware limitations, there are several methods to be considered for an improvement of the classification accuracy. First, other features could be used. For example, event-related desynchronization and event-related synchronization in alpha- and beta-bands, respectively, can be observed during the execution of motor imagery. Due to the non-stationary and transient nature of EEG signals, the discrete wavelet transform (DWT) is widely used for BCI systems. Another promising approach are common spatial patterns (CSP) as features. Dynamic time warping (DTW) could also be considered. Furthermore, deep convolutional neural networks (CNN) or recurrent neural networks (RNN) could significantly improve classification accuracy and therefore enable the system for a multiclass classification of EEG signals. Especially the different features could be evaluated without too much additional effort. However, due to time constraints this was not possible and could be done in future work.

List of Figures

Bibliography

[1] N. Birbaumer, "Brain–computer-interface research: coming of age," 2006.

[2] A. Sivakami and S. S. Devi, "Analysis of eeg for motor imagery based classification of hand activities,"

[3] J. Meng, S. Zhang, A. Bekyo, J. Olsoe, B. Baxter, and B. He, "Noninvasive electroencephalogram based control of a robotic arm for reach and grasp tasks," *Scientific Reports*, vol. 6, 2016.

[4] X. Yong and C. Menon, "Eeg classification of different imaginary movements within the same limb," *PloS one*, vol. 10, no. 4, p. e0121896, 2015.

[5] J. Decety, "The neurophysiological basis of motor imagery," *Behavioural brain research*, vol. 77, no. 1, pp. 45–52, 1996.

[6] M. Lotze, P. Montoya, M. Erb, E. Hülsmann, H. Flor, U. Klose, N. Birbaumer, and W. Grodd, "Activation of cortical and cerebellar motor areas during executed and imagined hand movements: an fmri study," *Journal of cognitive neuroscience*, vol. 11, no. 5, pp. 491–501, 1999.

[7] R. W. Homan, J. Herman, and P. Purdy, "Cerebral location of international 10–20 system electrode placement," *Electroencephalography and clinical neurophysiology*, vol. 66, no. 4, pp. 376–382, 1987.

[8] C. A. Kothe and S. Makeig, "Bcilab: a platform for brain–computer interface development," *Journal of neural engineering*, vol. 10, no. 5, p. 056014, 2013.

[9] Y. Renard, F. Lotte, G. Gibert, M. Congedo, E. Maby, V. Delannoy, O. Bertrand, and A. LÃ©cuyer, "Openvibe: An open-source software platform to design, test, and use brain-computer interfaces in real and virtual environments.," *Presence*, vol. 19, no. 1, pp. 35–53, 2010.

[10] G. Schalk and J. Mellinger, *A practical guide to brain–computer interfacing with BCI2000: General-purpose software for brain-computer interface research, data acquisition, stimulus presentation, and brain monitoring*. Springer Science & Business Media, 2010.

[11] I. Biopac Systems, "Biopac product site," 2017.

[12] S. Ge, R. Wang, and D. Yu, "Classification of four-class motor imagery employing single-channel electroencephalography," *PloS one*, vol. 9, no. 6, p. e98019, 2014.

[13] R. Leeb, F. Lee, C. Keinrath, R. Scherer, H. Bischof, and G. Pfurtscheller, "Brain–computer communication: motivation, aim, and impact of exploring a virtual apartment," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 15, no. 4, pp. 473–482, 2007.

[14] Z. Tayeb, E. Ercelik, and J. Conradt, "Decoding of motor imagery movements from eeg signals using spinnaker neuromorphic hardware," *arXiv*, 2017.

License