Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

Kjersti Brynestad and Erlend Vatsvåg

An Asynchronous Motor Imagery-based Brain-Computer Interface for Two-dimensional Drone Control

Master's thesis in Engineering Cybernetics
Supervisor: Marta Molinas
Co-supervisor: Luis Alfredo Moctezuma
June 2021

NTNU
Norwegian University of Science and Technology

Kjersti Brynestad and Erlend Vatsvåg

An Asynchronous Motor Imagery-based Brain-Computer Interface for Two-dimensional Drone Control

Master's thesis in Engineering Cybernetics
Supervisor: Marta Molinas
Co-supervisor: Luis Alfredo Moctezuma
June 2021

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

NTNU
Kunnskap for en bedre verden

Preface

This Master's thesis completes a Master of Technology at Norwegian University of Science and Thecnology (NTNU) under the Department of Engineering Cybernetics, carried out during the spring semester of 2021. The idea of the project was provided by Marta Molinas, who also acted as our supervisor. The work in this project was initiated with two specialization projects carried out during the fall semester of 2020, written by the two authors.

Our 2nd supervisor, Luis Alfredo Moctezuma, suggested and provided code for the Discrete Wavelet Transform (DWT) method, DWT-based feature extraction and Support Vector Machine (SVM), as well as skeleton code for the NSGA-II algorithm. Additionally, example code of a drone state machine and OpenBCI data acquisition developed by himself and previous students was provided and adapted to this system by us. Moreover, open-source code developed for the EEG headset by OpenBCI and an API for the drone developed by J. Philipp de Graff was used. The other methods implemented throughout this thesis are found in various sources, through literature search carried out by us, with support and suggestions from our 2nd supervisor.

Parts of Chapter 1, 2 and 3 are improved and extended versions of what was presented in our specialization projects [1, 2]. The system design presented in Chapter 4 is created by us. The datasets described and used for analysis are also recorded by us, with volunteering subjects. The results, discussion and conclusion presented in Chapter 5 and 6 are our original work.

The dataset can be shared upon request to one of the authors, at kjers.br@gmail.com or erlend_v1411@hotmail.com. A video of an online test with the drone can be found here (youtu.be/ZvIhkh20hwU).

We had no prior experience with Motor Imagery, EEG analysis or digital signal processing when starting this project fall 2020. The new knowledge and skills gained during this project is hopefully reflected in this thesis.

Trondheim, 2021-06-14

Kjersti Brynestad and Erlend Vatsvåg

Acknowledgement

We want to thank our supervisor Marta Molinas for introducing us to this interesting field and having belief in our ability to see the project through. We would also like to give a special thanks to our co-supervisor, Luis Alfredo Moctezuma, for always being available with help and guidance, both academically and logistically, and for providing and discussing ideas.

Particular gratitude is also given to the subjects who participated in the experiments. We are beyond thankful for the time they set aside and the enthusiasm they had for the project, as it would not have been possible without them.

We would also like to thank NTNU for access to the computing resources in the High Performance Computing system IDUN. This enabled the use of a multi-objective optimization algorithm in this work.

Finally, we want to thank our friends and family, for the useful discussions, continuous love, support and encouragement they have provided.

Abstract

This work investigates different state-of-the-art pre-processing techniques, feature extraction methods and classification algorithms for self-recorded Electroencephalography (EEG) Motor Imagery (MI) data, for the purpose of creating a Brain Computer Interface (BCI) capable of classifying MI tasks and convert it into commands to control a drone in real time.

As the field is still evolving, plenty of approaches have been suggested through the literature. To address this, a selection of state-of-the-art methods were tested using a Multi-objective Evolutionary Algorithm called Non-dominated Sorting Genetic Algorithm II, which was used to optimize methods for individual subjects. This work also proposed the use of a hierarchical structure for classification of MI tasks and resting-state, but found it inferior to the flat structure.

Out of 16 subjects, a BCI was implemented for the two highest-performing subjects, subjects 2 and 15. For offline classification of right hand MI, left hand MI and resting-state, subject 2 obtained a classification accuracy of 84.17% using Riemannian Geometry-based features and Logistic Regression (LR), while subject 15 obtained a 74.17% accuracy using Common Spatial Pattern (CSP) and Support Vector Machine (SVM). For right hand MI, left hand MI, foot MI and resting-state, subjects 2 and 15 obtained a classification accuracy of 76.88% and 67.50% respectively, both with Riemannian Geometry-based feature extraction methods.

Through extensive testing, the scheme using right hand MI, left hand MI, foot MI and resting-state was found infeasible for online classification with drone control using the obtained data, due to low accuracy. With right hand MI, left hand MI and resting-state, sufficient accuracy was obtained for a BCI with two-dimensional real-time control of the drone. This resulted in a true positive rate of 70.37% and false positive rate of 0% for subject 2, while the subject 15 managed to obtain a true positive rate of 91.17% and a false positive rate of 8.33%.

A successful system for controlling a drone with left and right hand MI was implemented, but for a limited number of subjects in controlled environments. The system had two control commands, a low throughput of commands and a high response time, and is therefore not yet applicable in practical settings.

Sammendrag

Dette arbeidet undersøker ulike state-of-the-art pre-prosesseringsmetoder, metoder for å ekstrahere egenskaper og klassifiseringsalgoritmer til bruk på Elektroencefalografi (EEG) data av forestilte bevegelser (MI). Datasettet som brukes lages som en del av dette arbeidet. Formålet er å lage et hjerne-datamaskin-grensesnitt (BCI) som kan klassifisere MI og konvertere det til kommandoer for å kontrollere en drone i sanntid.

Siden forskningsområdet er under utvikling, har flere fremgangsmåter for klassifisering blitt foreslått i litteraturen. Et utvalg av metoder ble derfor tested ved hjelp av en multi-objektiv evolusjonær optimaliseringsalgoritme, med mål om å finne de beste metodene for hver bruker. Dette arbeidet foreslo også en hierarkisk struktur i klassifiseringen, men denne strukturen oppnådde dårligere resultater enn en flat struktur.

En BCI ble implementert for de 2 forsøkspersonene som oppnådde best resultater, av totalt 16 personer. For offline klassifisering av høyre hånds-MI, venstre hånds-MI og en nøytral mental tilstand, oppnådde en forsøksperson en treffprosent på 84.17% ved bruk av Riemannsk geometri-baserte metoder, mens den andre forsøkspersonen oppnådde en treffprosent på 74.17% med en Common Spatial Pattern (CSP)-basert metode. For høyre hånds-MI, venstre hånds-MI, fot-MI og en nøytral mental tilstand, oppnådde de to samme forsøkspersonene en nøyaktighet på 76.88% og 67.50%, begge med Riemannsk geometri-baserte metoder.

Gjennom omfattende testing ble det konkludert med at det ikke var gjennomførbart med klassifisering i sanntid av høyre hånds-MI, venstre hånds-MI, fot-MI og en nøytral mental tilstand med de ervervede dataene. Med høyre hånds-MI, venstre hånds-MI og en nøytral mental tilstand derimot, var det mulig å implementere to-dimensjonal kontroll av dronen i sanntid. For det ene individet resulterte dette i en hyppighet av sanne positive prediksjoner på 70.37 %, og en hyppighet av falske positive prediksjoner på 0 %. For det andre individet, ble en hyppighet av sanne positive prediksjoner på 91.17% og falske positive prediksjoner på 8.33% oppnådd.

Et fungerende system for å kontrollere en drone med høyre hånds-MI og venstre hånds-MI ble implementert, men kun for to forsøkspersoner, og i et kontrollert forsøksmiljø. Systemet har relativt få frihetsgrader for kontroll med to mulige kommandoer, en lav gjennomstrømning av kommandoer og lang responstid. Det er derfor, enn så lenge, lite anvendelig i praktiske sammenhenger.

Contents

List of Figures

List of Tables

List of Abbreviations

API Application Programming Interface

BCI Brain-Computer Interface

CAR Common Average Reference

CNN Convolutional Neural Network

CSP Common Spatial Pattern

DWT Discrete Wavelet Transform

EA Evolutionary Algorithm

EEG Electroencephalography

ELU Exponential Linear Unit

EMD Empirical Mode Decomposition

EMG ElectroMyoGram

EOG ElectroOculoGram

ERD Event-Related Desynchronization

ERS Event-Related Synchronization

FBCSP Filter Bank Common Spatial Pattern

FFT Fast Fourier Transform

FGDA Fischer Geodesic Discriminant Analysis

FPR False Positive Rate

FT Fourier Transform

ICA Independent Component Analysis

LDA Linear Discriminant Analysis

LR Logistic Regression

MDRM Minimum Distance to the Riemannian Mean

MEMD Multivariate Empirical Mode Decomposition

MI Motor Imagery

MOOP Multiobjective Optimization Problem

NSGA-II Non-dominated Sorting Genetic Algorithm II

PCP Parallel Coordinate Plot

R-CSP adaption of CSP to the Riemannian manifold

SCM Sample Covariance Matrix

SNR Signal-to-noise ratio

SPD Symmetric Positive-Definite

SSVEP Steady-State Visual Evoked Potential

SVM Support Vector Machine

TPR True Positive Rate

TS Tangent Space

Chapter 1

Introduction

1.1 Background

A Brain-Computer Interface (BCI) provides the possibility to interact and communicate with external devices directly with the brain, without the use of muscles, peripheral nerves, or external controller devices [3]. By using sensor technology to measure brain activity, the brain signals can be interpreted with machine learning and converted into commands based on a pre-defined set of outputs. Electroencephalography (EEG) is a brain activity recording technique often used, as it is a non-invasive and therefore practical method. MI is a neuro-paradigm that can be used in BCIs, as it allows for consciously imagined limb movements to be converted to control output for e.g. a physical application [4].

MI-based BCIs have been subject to extensive research, as it has great potential in fields such as neurorehabilitation, neuroprosthetics and other assisting applications for the physically impaired [5]. This area of research could potentially let such patients control their environment directly with their brain. In addition to restoring or replacing bodily functions, BCIs can act as a *supplement* in the healthy. Some examples include gaming, monitoring the attentiveness of airline pilots [5], or, of course, actuating a drone, which is the objective of this work. As it is pointed out in [6], development of BCIs that act as supplements in the healthy can be important to fuel the progress towards commercializable BCIs restoring or replacing lost bodily functions. The expected profit is typically low for medical technology made for small groups of patients [6]. In non-medical BCIs made as a supplement for the average consumer or in profitable industries, however, the expected profit of commercialization could be higher, and could therefore attract the interest of the industry which could fuel the research towards commercialization with their resources.

Among the different types of BCIs, asynchronous, spontaneous and non-invasive BCIs are particularly interesting, as this would allow for self-paced control of the BCI without dependence on external stimuli. Unlike asynchronous BCIs, synchronous BCIs can only receive commands in specific time intervals, and cannot be operated at the user's pace [7]. Hence, the synonym for an asynchronous BCI, *self-paced* BCI. Furthermore, a *reactive* BCI depends on external stimuli, whereas an *active* BCIs depend only on the intention of the user to control their brain activity. The BCI being non-invasive means the user can easily inaugurate the system without undergoing surgery [4].

As these types of applications require reliable control before they could be put to use, the algorithms which analyse and classify the data have to be accurate and consistent. Even though there has been ongoing research on BCIs for the last two decades, the technologies are mostly restricted to research environments [5]. More research is necessary before MI-based BCIs using non-invasive EEG is reliable enough to use outside of controlled research environments.

There are several challenges left to solve in the research related to MI-based BCIs. They can all be summed up to revolve around increasing robustness, accuracy, efficiency and usability. Some of the most important challenges are to develop better sensors and improve signal processing techniques and classification methods [5]. Another challenge is to increase the accuracy of detection of the correct imagined movement with an increased number of different imagined movements, to expand the possibilities for control of a BCI.

Furthermore, asynchronous BCIs have typically not performed as good as synchronous BCIs [7]. Therefore, a big challenge is to increase the performance in this paradigm. Asynchronous BCIs introduce the requirement to handle the *no control intention state*, *resting state* or *idle state*, where the user does not perform MI and does not intend to send commands to the BCI [8]. It is important to minimize the number of commands detected when the user has no control intention [7], in addition to detecting the correct imagined movement.

Other challenges include making the systems usable in uncontrolled, noisy environments, reducing calibration time for new users and erasing need for re-calibration. Lastly, solving the problem that form, quality, consistency and amount of brain responses during MI vary from user to user, making the user an important factor in the performance of the system [5].

1.2 Objectives

The purpose of this work is to design and implement an MI-based BCI for controlling a drone. As part of this work, a complete system including a data collection protocol, signal processing and drone control should be designed and implemented. State-of-the-art methods should be investigated and adapted to the system. The system should be tested in real-time, if feasible.

The problem definition of implementing a BCI for controlling a drone through MI can be broken down to the following main objectives for this work:

1. Design and implement a protocol for recording MI data. The protocol should be intuitive and simple, such that users who are not familiar with MI or such systems can use it after a short introduction and demonstration.

2. Create datasets from volunteering subjects using the said protocol.

3. Research and implement appropriate existing pre-processing methods to enhance the Signal-to-noise ratio (SNR) in the obtained data.

4. Research state-of-the-art algorithms for feature extraction and classification and select a set of algorithms to test for distinguishing between each of the different imagined movements and resting-state in the obtained data.

5. Conclude on an algorithm that obtains a satisfactory accuracy for this data, ideally as high accuracy as possible. The algorithm must also work in real-time.

6. Design and implement a state machine for simple control of the drone, based on the predictions from the real-time classification.

7. Make it possible to operate the drone with little to no re-calibration right before the flight. In other words: The algorithm should make it possible to classify new data, based on training data that was recorded at least a week before for the same user.

1.3 Approach

The problem is approached as following. To identify possibly suitable methods for signal processing, feature extraction and classification, a review of state-of-the-art methods was conducted. Data from 16 volunteering subjects performing two different MI tasks was collected using the designed protocol and their performance was evaluated. The best-performing subjects were identified and invited to further experiments with three MI tasks.

A multi-objective genetic optimization algorithm was used to evaluate different state-of-the-art methods with respect to classification accuracy. In these experiments, different re-referencing methods and extraction of different frequency bands was tested. For feature extraction, mainly Common Spatial Pattern (CSP) and Riemannian Geometry-based methods were tested, as well as Discrete Wavelet Transform (DWT)-based features. In classification, Linear Discriminant Analysis (LDA), Support Vector Machine (SVM) and Logistic Regression (LR) were mainly used.

To identify possible approaches to the online classification and drone control, other work implementing online MI classification or designing BCI systems was reviewed. To verify our approach before online experiments, a simulation of an online experiment was conducted using the acquired data with two and three MI tasks. Lastly, a short experiment of controlling the drone online was carried out with the two best-performing subjects, using two MI classes, which allowed control in two dimensions.

1.4 Limitations

The main focus of this work is the methods for signal processing, feature extraction and classification. The scope of this work is limited to experimental results, meaning a deployable software is not developed. Furthermore, the hardware to be used was pre-determined, and issues considering hardware is therefore out of the scope of this work.

The experiments are only performed on the datasets recorded in this work. Due to limitations in time for the project and the availability of subjects, a limited amount of data could be recorded. Also, limited time was available for the final, online experiment.

1.5 Outline

This thesis is structured as following. In chapter 2, relevant background theory is explained, from the relevant brain functions, MI and BCIs, to methods for signal processing, feature extraction and classification. In chapter 3, a review of state-of-the-art in MI-based BCIs is given. Chapter 4 addresses the tools used, the procedure for data acquisition, a description of the obtained datasets, as well as the proposed BCI design. In chapter 5, experiments and results are presented. Lastly, the results are discussed in chapter 6, and a conclusion and recommendations for further work are given.

Chapter 2

Theoretical Background

The purpose of this chapter is to provide the theoretical basis for the methods used in this work. First, the relevant processes of the brain are described to show why it is possible to recognise specific processes. Then, to provide an introduction to *how* these processes can be recognised, theory about EEG and BCIs is introduced, as well as methods for signal processing, feature extraction and classification relevant for this work. Lastly, an algorithm used for optimizing methods and parameters is described.

2.1 The Human Brain

To design a BCI, it is important to understand the human brain, how it communicates, how this communication can be measured and interpreted, and which part of the brain is responsible for what. This will all be discussed in this section.

2.1.1 Brain Activity

The bodily and mental functions of the human are possible because of electrical and chemical signals in the body. The electrical signals are transmitted through interconnected neurons. A membrane potential, i.e. a difference in voltage between the inside and the outside of the nerve cell, is generated by controlling the concentration of ions on each sides of the membrane of the neurons. This negative electrical potential of a nerve cell in its ordinary state is called the resting potential. When a neuron is activated, the inflow and outflow of ions causes the membrane potential to depolarize. This event is called the action potential [9].

Brain activity stems from groups of neurons firing, causing electrical signals to travel across the brain. These electrical signals are essential for controlling bodily and mental processes. Different areas of the brain are related to different processes, which means that by looking at specific types of electrical activity in specific parts of the brain, one can get information about the processes happening in the brain [9].

2.1.2 Frequency Bands of the Brain

The electrical brain activity shows up as *brainwaves*, i.e. oscillatory activity. The frequencies of these oscillations are generally considered to be ranging from around 0.1 to 100 Hz. This range is often split up into a set of frequency bands, as presented in table 2.1 [4].

Table 2.1: Definition of the frequency bands of the brain from [4].

Frequency band	Frequencies
delta (δ)	< 4 Hz
theta (θ)	$4 - 7$ Hz
alpha (α)	$8 - 12$ Hz
beta (β)	$12 - 30$ Hz
gamma (γ)	> 30 Hz

Each frequency band is often associated with different mental states. The slower waves, such as the delta wave, are more common during the state of deep sleep, while gamma waves are more dominant during deep focus [10]. However, according to [11], these definitions vary and one should be careful mapping frequency bands directly to specific brain processes.

2.1.3 Motor Cortex

The human brain is divided into four main parts, *cerebral hemisphere, dienceohalon, cerebellum* and *brainstem*. The cerebral hemisphere, the largest part of the human brain, can be split into four different lobes with different functions, namely the *occipital, temporal, parietal* and *frontal* lobe [9]. Several parts of the brain are involved in motor control [9], but in relation to BCIs, the control of motor activity is most typically associated with the frontal lobe. More precisely, it is associated with the primary motor cortex, placed in the back of the frontal lobe [4]. This is illustrated in fig. 2.1.

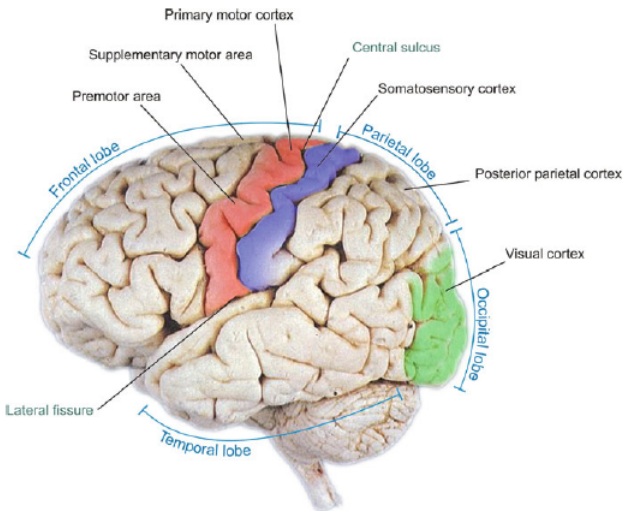


Figure 2.1: Overview of the occipital, temporal, parietal and frontal lobes. The position of the motor cortex within the frontal lobe is highlighted in red. Reprinted from figure 6 in [4].

In the primary motor cortex, different areas are associated with motor control of different limbs. Control of fingers and facial muscles require high precision, resulting in large areas being devoted to these limbs. Less precise motor control, such as arm or trunk, require less area of the motor cortex [12]. This mapping of a single hemisphere, i.e. a single half of the motor cortex, is illustrated in fig. 2.2. The motor control is lateralized, meaning that the left hemisphere controls the limbs of the right half of the body, while the right hemisphere controls the left side. To obtain a full cortical map of both hemispheres, one could simply mirror fig. 2.2.

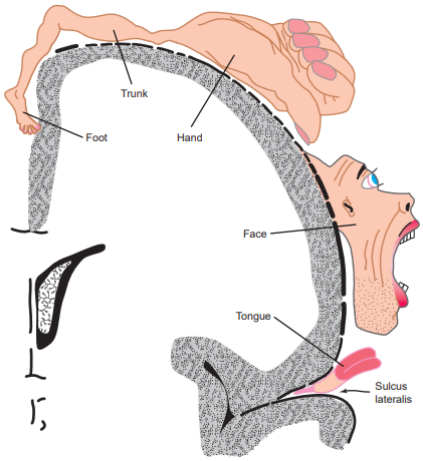


Figure 2.2: A map of the motor cortex - the motor cortical homunculus. Mapping of different limbs to different parts of the motor cortex in a single hemisphere. Reprinted from figure 28.6 in [12].

2.2 Electroencephalography

EEG is a technique used to record the brain activity generated by the neurons in the brain. The method is non-invasive, as no skin or mucous membranes are breached, making it relatively convenient to use. EEG uses electrodes directly on the scalp, which measures voltage potentials from local current flows that are produced by activation of large populations of neurons in the cerebral cortex [11]. Each electrode on the scalp records brain activity and is denoted as channels. There are limitations to how strong EEG signals are, as the signals from the neurons have to pass through skin, cerebral spinal fluid and brain volume to reach the electrodes. It is therefore also prone to distortion, environmental noise and artifacts [11].

2.2.1 Electrode Placement

The electrodes can be placed according to different schemes. The relevant scheme in this work is the international extended 10-20 system. In fig. 2.3, an illustration of the extended 10-20 system is shown [13]. In this system, the electrodes to use are chosen based on their placement and the goal of the recording. As different parts of the brain are mapped to specific tasks, this can be used to an advantage, as the correct position can result in desired brain recordings. Also, EEG require the use of a reference electrode, as all signals are measured as voltages relative to this reference. For this, one of the electrodes in the montage can be chosen, for instance electrodes A1 or A2 at the earlobes.

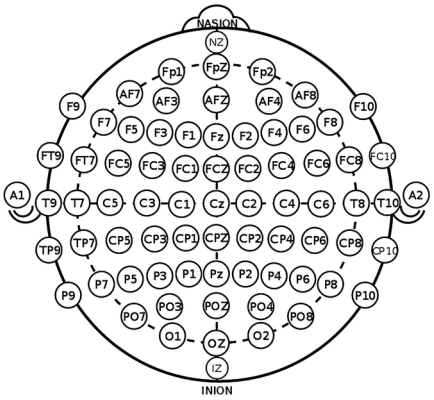


Figure 2.3: Electrode placement according to the extended 10-20 system. Illustrates the electrodes placed on a head from above, with the nose pointing upwards. Adapted from [13].

2.2.2 Artifacts in EEG

Artifacts are electrical activity that contaminates the EEG signal, and is unrelated to the activity in the brain. They are typically divided into *physiological* and *non-physiological* artifacts. Non-physiological artifacts have sources not related to the human body. This typically includes power-line noise, malfunction within the recording device or poor connection of the electrodes. Physiological artifacts include ocular artifacts, i.e. ElectroOculoGram (EOG) artifacts, which are related to eye movements and blinking. It also includes artifacts from muscle activity, i.e. ElectroMyoGram (EMG) artifacts, like jaw clenching, swallowing and talking, and scalp perspiration and movements [13]. Artifacts are primarily a problem in non-invasive EEG [4].

The impact of some artifacts can effectively be reduced by filtering, e.g. the 50 Hz (Europe) or 60 Hz (US) power-line noise, or low-frequented (~0.5 Hz) scalp perspiration artifact [13]. Other artifacts, especially EOG and EMG, have a wider frequency range overlapping with frequency bands of interest for analysis and are therefore more difficult to remove [14, 15]. Eye blink artifacts generally have a much higher amplitude than the regular EEG signal. EMG artifacts are more difficult to stereotype, but the shape and amplitude generally correlate with the degree of muscle contraction and which muscle is contracted [15].

2.3 Motor Imagery

MI is the mental process of imagining or thinking about body movement, without actually performing any muscular movement. MI can also be seen as the naturally and unconsciously process when preparing or intending a movement. In addition to being a widely used neuro-paradigm in BCIs, it is a process which has been investigated for performance improvements in both musicians and athletes, as the brain regions being activated during motor execution are also being activated during MI [16].

Performing MI induces specific changes in the brainwaves. These changes happen in the *mu* (μ) and *beta* (β) frequency brands. The mu band is alpha activity, i.e. 8 to 12 Hz, that is recorded from the

motor and somatosensory cortex. From table 2.1, it can be seen that beta activity ranges from 12 to 30 Hz. The changes that happen are, more specifically, Event-Related Desynchronization (ERD) and Event-Related Synchronization (ERS). ERD is an amplitude reduction or attenuation of certain frequency bands, while ERS is an amplitude enhancement [4].

Patterns of ERD and ERS can be voluntarily produced by performing imagined movements of different limbs. These patterns follow the cortical homunculus, meaning that imagination of movement of a specific limb, the *attended cortical body-part*, induces ERD in the mu band in the cortical area corresponding to that limb in fig. 2.2. According to [17], it has also been shown that MI can induce ERS of mu and beta rhythms in *non-attended cortical body-parts*, i.e. all other parts of the motor cortex except the one corresponding to the limb that is currently imagined. For example, imagined foot movement can induce ERS in the mu rhythm in the cortical area corresponding to hand movement. This behavior of ERD and ERS in attended and non-attended body-parts can be called *focal ERD/surround ERS*. Like in motor control, these responses are lateralized, see section 2.1.3. Some other changes that can occur are ERS of beta rhythm in attended cortical body-parts or ERD of central beta rhythm, as well as short-lasting beta ERS after the imagined movement ceases [17].

Because the ERD/ERS patterns are lateralized and follow the homuncular organization, electrodes can be placed according to the position of each limb in the motor cortical homunculus to map ERD/ERS patterns to imagined movements of different limbs. The four imagined movements most easily distinguished and mainly used in MI-based BCIs using EEG are right hand, left hand, arbitrary foot and tongue. As can be seen in fig. 2.2, these motor functions have a large corresponding area in the motor cortex. It is not possible to differentiate between left and right foot MI since they are located very close. Electrode C3 is considered most important to detect right hand MI, C4 for detecting left hand MI and Cz for foot MI [4].

MI can be executed both visually and kinesthetically. In visual MI, one attempts to imagine what the movement looks like, while kinesthetic imagery is the act of trying to *feel* the movement [18]. Imagined movements of different limbs are denoted as *MI tasks* or *MI classes* in this work.

The amount of brain responses elicited during MI is very different from user to user, and even within a user over time. For most users, training with feedback is required to learn to induce sufficient ERD/ERS patterns [19]. However, a big portion of users, an estimated 15-30 %, are so-called *MI-based BCI illiterates*, and do not show enough of the expected ERD/ERS pattern to control a MI-based BCI accurately, even after training. This is a major problem in MI-based BCIs [19].

2.4 Brain-Computer Interfaces

A BCI provides a user with a non-muscular control and communication channel that can convey messages and commands to external devices [20]. It offers an alternative to control through bodily movements, as it bypasses the normal physiological pathways of the body [4].

BCIs can be passive, reactive and active. A passive BCI derives outputs based on involuntary brain ac-

tivity, such as user state. A reactive BCI, also called an evoked BCI, produces outputs based on brain activity evoked by external stimuli. Active BCIs, also called spontaneous BCIs, derives output based on consciously controlled brain activity [21]. However, in [4], it is stated that devices that detects changes in brain activity that occurs without intent, are not BCIs. MI-based BCIs are of the spontaneous type, as they base the output on consciously controlled brain activity.

BCIs can either be synchronous or asynchronous. Synchronous BCIs rely on specific time intervals where the user can perform commands, whereas asynchronous BCIs can receive commands at any given moment [7]. As commands can be executed at any moment, asynchronous BCIs also need an idle-state, where no commands are given. This offers intuitive control for gaming and prosthetic [22]. For the idle-state, subjects may either relax entirely, trying not to think of anything, or do another mental task than the ones for the active classes [22].

Due to the explicitness of ERD/ERS patterns varying from user to user, as mentioned in section 2.3, the performance of a BCI is highly dependent on the user. However, BCI illiteracy is dependent on the underlying neuro-paradigm, so users who are MI-based BCI illiterates may well be able to use a BCI with a different underlying neuro-paradigm.

2.5 EEG Recording Protocols for MI

In order to record specific EEG events, a *recording protocol* can be used. A recording protocol consists of a set of cues given to a subject, indicating tasks to perform, according to predefined time windows. In the case of MI recording protocols, cues define when to perform MI and which movements to imagine. Between tasks, subjects are typically instructed to rest. It is also possible to add feedback to the user.

In MI protocols, both visual and auditory cues can be used. Sometimes both are used in combination, such as in *BCI Competition I, dataset 2b* [23]. Here, two MI classes, left and right hand, was performed. For this, a fixation cross was set in the middle of the screen. After two seconds, an acoustic warning tone would message the subject that an upcoming event was approaching. One second later, a visual cue presented as an arrow, would suggest which MI task to perform. The cue would last for 1.25 seconds, and the subject was to perform the MI for four seconds. In [24], a similar approach was used for 4 MI classes, only using visual cues. In fig. 2.4, an illustration of the timing of such protocols is given.

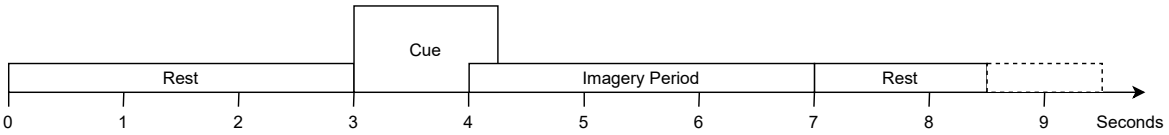


Figure 2.4: Example of a basic protocol for recording MI EEG data. Adapted from [23]

2.6 Pre-Processing

Raw EEG data can contain a large amount of noise, and therefore have a low SNR. As mentioned, EEG signals are also prone to artifacts. In addition, there are frequency bands which contain more useful information than others, and should therefore be isolated. For these reasons, different pre-processing techniques are implemented to enhance the signals SNR and increase the likelihood of good classification results [13].

As this thesis is focused on real-time applications of BCIs, all pre-processing is performed on segments of data instead of entire recording sessions. When implementing the real-time application, only a segment of data is available at each time step. Therefore, this is also done during training of classifiers, as all data has to be handled in the same manner. These segments are typically called epochs or trials, and all contain the same pre-decided number of samples.

2.6.1 High, Low and Band-pass Filter

To attenuate or pass certain frequencies, either high, low or band pass filters can be used. A high pass filter is a filter that attenuates frequencies below a given frequency, and lets the higher frequencies pass. A low pass filter is the opposite, as it attenuates the frequencies higher than a given frequency, and lets the lower frequencies pass. A band pass filter serves as a combination of the two, as it lets a range of frequencies pass, while attenuating the signals outside this range.

2.6.2 Notch Filter

As power-line noise is prominent in EEG signals, a notch filter can be implemented to filter out this noise. A notch filter works as a band stop filter with a very narrow stop-band. As opposed to a band pass filter, it suppresses the data of desired frequencies. As the power-line noise is of 50 Hz in Europe, a 50 Hz notch filter is used.

2.6.3 Re-referencing

What we want to record in each electrode location is the electrical activity from that exact location, without contamination from noise or other electrodes. Therefore, the choice of reference electrode plays an important part in EEG. The ideal reference electrode is placed in an electrically neutral location, such that it keeps a constant voltage. Otherwise, if it picks up some electrical activity, this activity will influence the signal recorded in the other electrodes. However, the conductivity of the scalp which is crucial for recording of brain signals on the scalp, also causes the problem that there is no such thing as an electrically neutral location on the head [25].

Since it is not possible to make the reference electrode completely independent of the other electrodes by e.g. placing it at an infinite distance from the source, the signal recorded from the other electrodes will be dependent on the location of the reference electrode or the method of referencing [25]. Therefore, it is in some cases appropriate to create a synthetic reference during signal processing, as it could enhance the SNR. According to [26], the two most commonly used methods for re-referencing in EEG are Common Average Reference (CAR) and Laplacian re-referencing. Studies have

found CAR and Laplacian re-referencing to be superior to the ear-reference method with respect to classification accuracy or at reflecting the sensorimotor activity [26, 27, 28]. Both methods are best suited for EEG data with focal activity [25], like, for instance, ERD and ERS related to MI.

Common Average Reference

CAR is an example of a global average reference, which means that all electrodes contribute to the reference [25]. Here, the reference is computed as the average of the activity in all electrodes. The CAR is computed using the following formula [28]:

$$V_i^{CAR} = V_i^{ER} - \frac{1}{n}\sum_{j=1}^{n} V_j^{ER} \tag{2.1}$$

where V_i^{ER} is the potential, i.e. the voltage, between the i^{th} electrode and the original reference, while n is the number of electrodes.

Laplacian Reference

When using Laplacian re-referencing, a unique reference is computed for each electrode. Thus, it is an example of a local average reference, which helps in enhancing localized activity [25]. When computing the Laplacian for a certain electrode, the surrounding electrodes are weighted according to their distance to the electrode currently being considered. The second derivative of the instantaneous spatial potential distribution is estimated by subtracting the sum of weighted voltages of surrounding electrodes. Specifically, the Laplacian montage is computed using the formula [28]:

$$V_i^{LAP} = V_i^{ER} - \sum_{j \in Si} g_{ij} V_j^{ER} \tag{2.2}$$

where

$$g_{ij} = \frac{\frac{1}{d_{ij}}}{\sum_{j \in Si} \frac{1}{d_{ij}}} \tag{2.3}$$

and the set *Si* is the surrounding electrodes of the i^{th} electrode. d_{ij} is the distance between the electrodes *i* and *j*. Definitions of this distance can vary, e.g. depending on the EEG headset in use. In this work, d_{ij} is derived from considering the electrodes as nodes in fully connected graph where all edges have length 1, as an approximation for the physical distances.

2.7 Fast Fourier Transform

In analysis of oscillating signals, such as brainwaves, the signal's frequencies and their relative power can be useful. One way to transform a signal from the time-domain to the frequency domain, is by using the Fourier Transform (FT). This conversion is executed through the mathematical formulation [29]:

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t}\, dt \tag{2.4}$$

This function can be transformed into a discrete function, Discrete Fourier Transform (DFT), by approximation to a finite sum. It can be formulated as [29]:

$$F_k = \sum_{n=0}^{N-1} f_k e^{\frac{-j2\pi kn}{N}} \tag{2.5}$$

By avoiding the repeated calculations that are occurring in the DFT, the caluclations can be reduced from N^2 to $NlogN$ calculations, resulting in the Fast Fourier Transform (FFT) [29]. As the method transforms a signal from the time-domain to the frequency-domain, it results in a loss of time characteristics. The method is still a powerful tool in digital signal analysis.

2.8 Discrete Wavelet Transform

The DWT is a method for signal decomposition, representation and compression. The idea behind the transform is very similar to the FT, except instead of using sine and cosine as the basis function, a *wavelet* is used. A wavelet is a short, finite oscillation with a characteristic shape. There exists a number of different wavelet families, which means that unlike the FT, DWT refers to a set of transforms, not just a single transform [30].

Furthermore, the DWT has several advantages over the FT. The DWT gives information about both frequency and time, not just about frequency, because the use of finite wavelets makes the DWT localized in time. Moreover, it also gives a good resolution for different frequencies, as the window size is implicitly adapted to the frequency by scaling of the wavelet [30].

In short, the signal is compared to a basis function, the mother wavelet, at different scales and translations. This results in the signal being decomposed into a high- and low-frequency part in a predetermined number of *levels*. The output of the transform is a set of frequency sub-bands [30]. For instance, in the case of the sampling rate being 250 Hz, 4 levels of decomposition gives the decomposition shown in table 2.2.

Table 2.2: The frequency sub-bands following the use of four levels of decomposition with a signal of 250 Hz.

4 levels of decomposition	
Sub-band	**Freq. band [Hz]**
Detail coefficients, level 1 (D1)	62.5 - 125
Detail coefficients, level 2 (D2)	31.25 - 62.5
Detail coefficients, level 3 (D3)	15.625 - 31.25
Detail coefficients, level 4 (D4)	7.81 - 15.625
Approximation coefficients, level 4 (A4)	0-7.81

2.8.1 DWT-based Feature Extraction

After decomposing a signal with DWT or similar techniques, different features can be extracted from the decomposed signal. As this thesis is based on EEG signals, which are oscillatory signals, energy

based features can be helpful in characterizing the oscillations and possible ERS/ERD in selected levels of the DWT. For this thesis, two energy features are used to help differentiate between the MI classes. The features are extracted from chosen levels from the DWT decomposition, and are explained below.

Instantaneous Energy

The first energy feature extracted from DWT decomposed signals is Instantaneous Energy. This feature is extracted by taking the log of the average of every value squared from each level of decomposition. This helps reflect on the amplitude of a signal within each level of decomposition. Mathematically, it is calculated as follows [31]:

$$f_{IE} = log_{10}\left[\frac{1}{N_j}\sum_{r=1}^{N_j}[w_j(r)]^2\right] \tag{2.6}$$

where $w_j(r)$ is the wavelet coefficient at time r and band j, and $N_j = N/2^j$ samples.

Teager Energy

The second feature extracted from the DWT composed signal is the Teager energy. This feature reflects the variations in amplitude and frequency, and decreases noise, making it a robust parameter [32]. It is computed as shown below [31, 33]:

$$f_{TE} = log_{10}\left[\frac{1}{N_j}\sum_{r=1}^{N_j-1}\left(w_j^2 - w_j(r-1)w_j(r+1)\right)\right] \tag{2.7}$$

where $w_j(r)$ is the wavelet coefficient at time r and band j, and $N_j = N/2^j$ samples.

2.9 Common Spatial Patterns

The Common Spatial Pattern (CSP) is an algorithm which can calculate spatial filters for detection of ERD and ERS [34]. The algorithm calculates a filter matrix $\mathbf{W}$. In this filter, every column of $\mathbf{W}$ is denoted as a spatial filter, and each column of $\mathbf{W^{-1}}$ is denoted as a spatial pattern. For a signal $\mathbf{X} \in \mathbb{R}^{N_{ch}\times N}$, where N_{ch} is the number of channels and N is the number of samples, the algorithm creates a $\mathbf{W} \in \mathbb{R}^{N_{ch}\times N_{ch}}$. The filtered signal is then represented as [35, 36, 37]:

$$\mathbf{Z} = \mathbf{W}^T\mathbf{X}, \qquad \mathbf{Z} \in \mathbb{R}^{N_{ch}\times N} \tag{2.8}$$

$\mathbf{W}$ is created in such a way that the filtered signal $\mathbf{Z}$ maximizes the difference in variance between the classes of data [34]. The calculation of the filter follows these steps [35, 36]:

1. Calculate the respective covariance matrices, $\mathbf{R_a}$ and $\mathbf{R_b}$, for the two classes, where:

$$\mathbf{R} = \frac{1}{N}\sum_{n=1}^{N}\frac{\mathbf{X^n X^{n^T}}}{Tr(\mathbf{X^n X^{n^T}})}$$

2. Combine the two matrices, $\mathbf{R_c} = \mathbf{R_a} + \mathbf{R_b}$ and perform eigen decomposition. Construct a whitening matrix P:

$$\mathbf{P} = \sqrt{\Lambda_\mathbf{c}^{-1}}\mathbf{U_c^T}$$

3. The respective covariance matrices, $\mathbf{R_a}$ and $\mathbf{R_b}$, are now whitened into $\mathbf{S_a}$ and $\mathbf{S_b}$ as following:

$$S = \mathbf{PRP^T}$$

4. The eigen decomposition of $\mathbf{S}$ provides the equation

$$S = \mathbf{B}\Phi\mathbf{B^T}$$

where Φ is the matrix of eigenvalues in descending order. Matrix $\mathbf{B}$ is used to optimize variation between the classes, by:

$$\mathbf{V} = \mathbf{B^T P}$$

5. From $\mathbf{V}$, a select amount of filters are chosen to obtain the spatial filter $\mathbf{W}$

The method is dedicated to the oscillatory activity and band-power features [38]. It is originally defined for binary paradigms, as it maximizes variance between two classes [35, 36]. However, there are several extensions of CSP to better optimize for multiple classes, such as presented in [37] and [36]. In this project, an implementation of [36] has been used.

2.10 Classification with Machine Learning

For classification of data, different machine learning algorithms are used. In general, machine learning is a generic expression used for algorithms that learns or finds patterns from data. From the learned patterns, the computer may now make predictions on unseen data. The data that the algorithm learns from are also often labeled. In such situations, it is known as supervised learning [39]. In all cases of machine learning, the quality and size of the data are crucial to obtain a successful predictor. Therefore, machine learning is closely related to data analysis and statistics, and often uses ideas from probability and optimization [40].

In the subsequent sessions, different machine learning algorithms used in this thesis are explained.

2.10.1 Support Vector Machine

SVM is a supervised machine learning technique, which is mostly used for data classification. For SVM, the decision boundary hyperplane ideally separates the input data of different classes, thus being able to distinguish between them. The hyperplane is chosen based on the margin and the number of errors. The optimal hyperplane minimizes the number of errors, while it maximizes the margin, thus making it a trade-off between the two. The number of errors is the number of instances which are on the incorrect side of the hyperplane, while the margin is the smallest distance between the instances and the hyperplane [41, 42].

In cases where classes overlap, a linear soft margin that introduces a penalty parameter, allowing misclassified data to be used. In addition, the kernel parameter can be changed to allow projection of data in higher dimension. This can creates a nonlinear separation boundary [41, 42]. However, in this work, only a linear kernel is used.

2.10.2 Linear Discriminant Analysis

LDA is a commonly used technique for data classification, while also having diemsionality reduction properties [43]. The method guarantees maximum separability by maximizing the ratio of between-class variance to the within-class variance [43]. Easily put, the method finds a projection hyperplane, where the variance within each class is minimized, and the distance between the classes' means are maximized [44].

The algorithm was originally designed for binary classification, and the methods used are as following [44]:

- Calculate the sample means from each set:

$$\bar{x}_i = \frac{1}{N_i} \sum_{x \in i} x \tag{2.9}$$

  where i is the class, and N_i is the number of samples in the class.

- Calculate the scatter matrices:

$$S_i = \sum_{x \in i} (x - \bar{x}_i)(x - \bar{x}_i)^T \tag{2.10}$$

- Find a hyperplane defined by a vector, ϕ, that ideally minimizes the data sample variance. This can be expressed as:

$$\min_{\phi} (\phi^T S \phi) \tag{2.11}$$

  where $S = \sum_{cl=1}^{i} S_i$.

- Calculate scatter matrix between the classes:

$$S_{1,2} = (\bar{x}_1 - \bar{x}_2)(\bar{x}_1 - \bar{x}_2)^T. \tag{2.12}$$

- The objective is to find a hyperplane that maximizes distance between the means of the classes, while also minimizing the variance or scatter within each class:

$$\max_{\phi} \frac{\phi^T S_{1,2} \phi}{\phi^T S \phi} \tag{2.13}$$

For multiclass classification, the scatter matrices must be redefined and the linear transformation, ϕ, is obtained differently [44]:

- Intra-class matrix:

$$S = S_1 + S_2 + ... + S_n \tag{2.14}$$

- Inter-class matrix:

$$S_{1,..,n} = \sum_{i=1}^{n} p_i(\bar{x}_i - \bar{x})(\bar{x}_i - \bar{x})^T \tag{2.15}$$

  where p_i is the number of samples within the class, $\bar{x}_i$ is the mean for each class and $\bar{x}$ is the total mean vector.

- Obtain ϕ by solving the eigen value problem:

$$S_{1,...,n}\phi = \lambda S\phi \tag{2.16}$$

2.10.3 Logistic Regression

LR is a classifier which is based on probability. It uses the logistic function, also called *Sigmoid function*, as a cost function, which is mathematically formulated as [45]:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{2.17}$$

With this function, $\sigma(z)$ ranges from 0 to 1. The cost function is minimized for the training instances to the optimal decision boundary. Then, the classifier gives a probability for each of the inputs, based on the Sigmoid function.

2.10.4 Convolutional Neural Network

A Convolutional Neural Network (CNN) is one type of deep learning algorithm, which uses a back-propagation algorithm to discover different structures in large datasets. By utilizing this, the machine can change internal parameters in the structure of multiple layers, thus being able to learn representation of data on multiple levels of abstraction [46]. CNNs have obtained great success in pattern recognition tasks, such as image and voice recognition, during the past decade [47]. CNNs consist of an input layer, hidden layers and an output layer.

The most prominent layer used in CNNs, is the convolutional layer. It uses learnable kernels, which can be small in spatial dimensionality, but spreads along the input's depth. The convolutional layer will produce a 2D activation map, as each layer will convolve the filters across the spatial dimensionality [48].

Convolutional layers can be optimized by changing the size and stride of the filter. A good size will reduce the number of parameters and thus computational complexity, while also retaining the capabilities of pattern recognition. The strides will, together with the size, decide the amount of overlapping and output dimensions [48].

In addition to the convolutional layer, the CNN used in this thesis also uses the average pooling layer. Pooling layers reduces the dimensionality of data, which reduces the number of parameters and thus computational complexity [48]. Average Pool layers takes the mean of the elements in a pooling re-

gion [49]. For activation function, Exponential Linear Unit (ELU) is used, and is defined as [50]:

$$f(x) = \begin{cases} x, & x \geq 0 \\ a(e^x - 1), & x < 0 \end{cases} \tag{2.18}$$

A flatten layer is used to transform the data into a flat shape, before a dense layer is used, to give the desired amount of output nodes. Here, a softmax activation function is used. As the nature of the function returns a value between 0 and 1, it can be treated as a probability, and therefore be used for classification for multiple classes. The softmax function can be expressed as [50]:

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}, j = 1, ..., K \tag{2.19}$$

2.11 Riemannian Geometry-based Feature Extraction and Classification

Compared to methods like CSP with LDA, Riemannian Geometry-based methods in BCIs is a relatively new concept attracting increasing attention [38]. Because Riemannian Geometry-based methods can span from feature extraction and spatial filtering to classification algorithms, these methods are presented together, separately from the other methods on feature extraction and classification. Since the concept is not as well-established as the previously described methods, it is described in more detail.

A Riemannian manifold is a real and smooth manifold, where the tangent space at every point is a finite-dimensional Euclidean space, see the illustration in fig. 2.5.

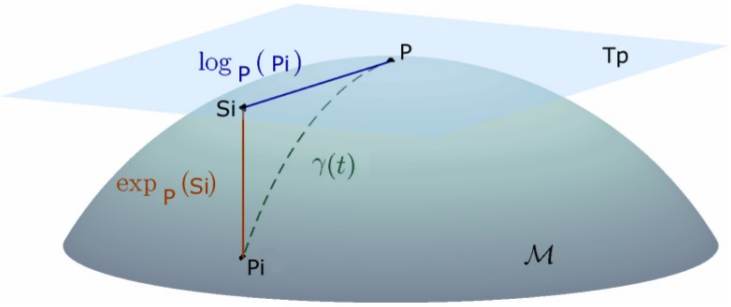


Figure 2.5: Illustration of the TS of manifold $\mathcal{M}$ at point $\mathbf{P}$, the tangent vector $\mathbf{S}_i$ at $\mathbf{P}$ and the geodesic $\gamma(t)$ between points $\mathbf{P}$ and $\mathbf{P}_i$. Reprinted from [51].

The shortest path in the Riemannian manifold between two points is called the *geodesic*, and is given by [51]:

$$\gamma(t) = \mathbf{P}_1^{1/2} \left(\mathbf{P}_1^{-1/2} \mathbf{P}_2 \mathbf{P}_1^{-1/2}\right)^t \mathbf{P}_1^{1/2} \tag{2.20}$$

The length of this path is the Riemannian distance, $\delta_R(\mathbf{P}_1, \mathbf{P}_2)$ [51]:

$$\delta_R(\mathbf{P}_1, \mathbf{P}_2) = ||\text{Log}(\mathbf{P}_1^{-1}\mathbf{P}_2)||_F = \left[\sum_{i=1}^{n} \log^2 \lambda_i\right]^{1/2} \tag{2.21}$$

between the two points $\mathbf{P}_1$ and $\mathbf{P}_2$. The Riemannian mean of m covariance matrices $\mathbf{P}_1, ...\mathbf{P}_m$ is defined as [51]:

$$\mathfrak{G}(\mathbf{P}_1, ...\mathbf{P}_m) = \underset{\mathbf{P}\in P(n)}{\text{argmin}} \sum_{i=1}^{m} \delta_R^2(\mathbf{P}, \mathbf{P}_i) \tag{2.22}$$

When using Riemannian Geometry-based methods, the EEG epochs are treated as points in the Riemannian manifold, represented by their covariance matrix. The idea is that related trials will lie close in this space. By classical methods such as CSP, covariance matrices are treated as if they were lying in the Euclidean space. However, they naturally belong in the space of Symmetric Positive-Definite (SPD) matrices, in which the native geometry is the Riemannian Geometry [51]. By Riemannian Geometry-based methods, the curvature of the space of SPD matrices is considered when the covariance matrices are analyzed [52]. An assumption that these methods rely on is that for a given mental state, the spatial distribution and power of the EEG sources is approximately fixed [38]. Then, this information of spatial distribution and power is contained in the analysis of the covariance matrices.

Different Riemannian geometry-based and Euclidean-based methods can be combined to a set of transforms to be performed on the data. First, the one mandatory feature transform and two optional feature transforms are presented. Then, the possible combinations of the methods are described, i.e. how to go from the mandatory transform to classification in the Euclidean space or in the Riemannian manifold, and when the optional transforms can be used, to give a picture of which transforms can be used together in which order.

2.11.1 The Covariance Matrix Estimate

Naturally, the covariance estimate is the mandatory, first transform in the Riemannian Geometry-based methods. The spatial covariance matrix can be estimated by different estimators. Some estimators, for instance the Sample Covariance Matrix (SCM), which is used as the default estimator in the original proposal of the Riemannian Geometry-based classifiers [51, 52, 53], are sensitive to outliers. Therefore, the use of estimates with regularization can improve the estimate [53]. In this work, the estimate used is the shrunk (regularized) Ledoit-Wolf covariance matrix, which was proposed keep the accuracy of the SCM, and unlike the SCM it would not amplify estimation error when inverted - in other words, it is *well-conditioned*. In the end, the estimate actually *improved* the accuracy from SCM while being well-conditioned [54].

To compute this estimate, a well-conditioned structured estimator is first computed by the condition that all covariances are zero and all variances are equal. Then, the SCM is computed [54]. Let $\mathbf{X}_p \in R^{E\times T}$ be a trial with number of channels E and number of samples T. The $E \times E$ SCM is: $\mathbf{C}_p = \frac{1}{T-1}\mathbf{X}_p\mathbf{X}_p^T$ [53]. Then, the estimator is a weighted average of the structured estimator and the SCM [54].

These weights are chosen according to a quadratic loss function, which ensures that this estimate is more accurate than either of the structured estimator and the SCM. The true optimal weights are defined from the minimization of the expected quadratic loss $E[||\sum^* - \sum||^2]$, where $\sum$ is the true, unknown covariance matrix. The goal is to find the linear combination $\sum^* = \rho_1 I + \rho_2 \mathbf{C}_p$, where I is the identity matrix and $\mathbf{C}_p$ is the SCM. However, this makes the estimate dependent on the true covariance matrix, $\sum$, which is unobservable. Therefore, an estimate of $\sum^*$ is computed, and it is proven that it has the same properties as $\sum^*$ [54].

2.11.2 Tangent Space Projection

After the covariance matrix estimate is computed, the next step can be to compute the Tangent Space (TS) projection. This operation projects the covariance matrices to the TS of the Riemannian manifold. Again, refer to fig. 2.5 for an illustration of the TS. This space is an Euclidean manifold, which means that the covariance matrix estimates after this operation can be treated as vectors in the Euclidean space, e.g. by applying the arithmetic mean. The TS projection is used in the case where it is desired to apply standard, state-of-the-art, Euclidean-based classification methods, but to simultaneously conserve the structure of the manifold, i.e. the spatial information, in the handling of the covariance matrix [52].

Let $\mathbf{P}$ be one point in the Riemannian manifold, and consider the set of other points $\mathbf{P}_i \in P(n)$. The tangent vector $\mathbf{S}_i$ for each point $\mathbf{P}_i$ is then identified by $\mathbf{S}_i = \dot{\gamma}(0)$, where γ is the geodesic between points $\mathbf{P}$ and $\mathbf{P}_i$, see eq. (2.20) and fig. 2.5. Then, the mapping from the Riemannian manifold to the TS, namely the *Riemannian Log map operator*, is given by [51]:

$$\mathrm{Log}_{\mathbf{P}}(\mathbf{P}_i) = \mathbf{P}^{1/2}\mathrm{Log}\left(\mathbf{P}^{-1/2}\mathbf{P}_i\mathbf{P}^{-1/2}\right)\mathbf{P}^{1/2} \tag{2.23}$$

While the inverse operation, the mapping from the TS to the Riemannian manifold, namely the *Riemannian Exp map operator*, is given by [51]:

$$\mathrm{Exp}_{\mathbf{P}}(\mathbf{S}_i) = \mathbf{P}^{1/2}\mathrm{Exp}\left(\mathbf{P}^{-1/2}\mathbf{S}_i\mathbf{P}^{-1/2}\right)\mathbf{P}^{1/2} \tag{2.24}$$

2.11.3 Filtering by Fisher Geodesic Discriminant Analysis

Fischer Geodesic Discriminant Analysis (FGDA) is a supervised filter algorithm which also optionally can be used as the next step after the covariance estimate has been computed. FGDA is an extension of Fisher's Linear Discriminant Analysis (LDA) to the TS. It is used to filter out information that increases the distance between covariance matrices from trials belonging to the same class, i.e. non-class-related information [51].

With FGDA, the data is projected to the TS, then LDA is applied to filter, before the filtered data is projected back to the manifold. More specifically [51]:

First, the filters are computed. Given the set of covariance matrices to filter $\mathbf{P}_i \in P(n)$,

1. Compute $\mathbf{P}_\Omega = \mathfrak{G}(\mathbf{P}_1, ...\mathbf{P}_n)$, the Riemannian mean of the set.
2. for all $\mathbf{P}_i \in P(n)$, apply Riemannian Log map with respect to $\mathbf{P}_\Omega$ (project to the TS in the point $\mathbf{P}_\Omega$) by the equation $\mathbf{S}_i = Log_{\mathbf{P}_\Omega}(\mathbf{P}_i)$. Vectorize $\mathbf{S}_i$: $\tilde{\mathbf{S}}_i = \text{vec}(\mathbf{S}_i)$
3. Compute filters $\tilde{\mathbf{W}} = \text{LDA}(\tilde{\mathbf{S}}_i)$
4. Select K components, i.e. K first vectors $\tilde{\mathbf{W}}_k$

Then, the filters are applied as following:

1. Apply Riemannian Log map, i.e. project to the TS $\mathbf{S}_i = \text{Log}_{\mathbf{P}_\Omega}(\mathbf{P}_i)$
2. Filter by $\tilde{\mathbf{S}}_i = \tilde{\mathbf{W}}\left(\tilde{\mathbf{W}}^T\tilde{\mathbf{W}}\right)^{-1}\tilde{\mathbf{W}}^T\text{vec}(\mathbf{S}_i)$
3. Then, the filtered covariance matrix is $\tilde{\mathbf{P}}_i = \text{Exp}_{\mathbf{P}_\Omega}\left(\text{unvec}(\tilde{\mathbf{S}}_i)\right)$

2.11.4 From Covariance Matrix to Classification in the Euclidean Space

When the covariance matrix has been obtained and one want to apply standard, state-of-the-art Euclidean-based classification methods while interpreting the covariance matrices in the native geometry, there are two options: Projecting the covariance matrices to the TS as described in section 2.11.2, or using a modified version of CSP that takes covariance matrices as input as described below.

Before describing the modified CSP, it is worth to mention that in between the calculation of the covariance estimate and either of the two transforms to the Euclidean space, FGDA filtering can be applied. However, note that applying FGDA before projection to the TS will include an excess projection from the TS to the manifold and back: If we denote the operation of projecting data from the TS back to the manifold as Inverse TS (ITS), we can describe FGDA as $TS \rightarrow LDA \rightarrow ITS$. Then, applying FGDA before applying the TS projection before classification in the Euclidean space would correspond to $TS \rightarrow LDA \rightarrow ITS \rightarrow TS$ which further corresponds to $TS \rightarrow LDA$. This does however *not* mean that applying the FGDA and then TS is the same as applying TS and a standard implementation of LDA, as the LDA classifier only uses one component K. Therefore, the two operations would only be equal if one component was also used in the computation of the filters in FGDA, which is not necessarily the case [51].

Modified CSP with Riemannian Metric

In [55], a modified version of CSP is proposed, an adaption of CSP to the Riemannian manifold (R-CSP). This version takes the previously computed covariance matrix as input, rather than incorporating the computation of the covariance matrices into the algorithm. Moreover, instead of using the arithmetic mean, the Riemannian mean is used [55]. This is implemented by changing step 1 of the CSP algorithm as described in section 2.9, where the mean of the covariance matrices is computed for each class. Also, step 5., i.e. choosing the final filters, is modified by utilizing the Riemannian distance to choose the filters. This is done because the distance between class means can be considered as the most important factor to differentiate two classes. The concept in CSP is to maximize the separability of the classes in the feature space, so in this case, spatial filters are selected based on the highest

Riemannian distance between the mean covariances. Lastly, a log-variance feature extraction is also performed on the spatially filtered signal.

Consider the filtered signal $\mathbf{Z} = [\mathbf{z}_1 ... \mathbf{z}_J]^T$, see eq. (2.8) in section 2.9. Then, the log-variance features are defined as [55]:

$$\mathbf{F_X} = \begin{pmatrix} \log(\text{Var}(\mathbf{z}_1)) \\ ... \\ \log(\text{Var}(\mathbf{z}_J)) \end{pmatrix} \tag{2.25}$$

The output of R-CSP can now be used in the Euclidean space with standard vector-based methods. Again, multi-class compatibility is achieved according to the definition in [36].

2.11.5 From Covariance Matrix to Classification Natively in the Space of SPD Matrices

Instead of moving to the Euclidean space for the classification step, the classification can also be performed entirely in the Riemannian manifold. For this, the algorithm Minimum Distance to the Riemannian Mean (MDRM) can be used.

Riemannian Minimum Distance to the Mean

MDRM is similar to the Minimum Distance to the Mean (MDM) algorithm, but instead of standard Euclidean distance and mean, Riemannian distance and Riemannian mean is used. See the definitions of Riemannian distance and Riemannian mean in eq. (2.21) and eq. (2.22), respectively. MDM is a simple classification algorithm, where the mean is computed for each class from known trials during training, while during prediction, the unknown trial is assigned to the class which has the closest mean.

2.12 Non-dominated Sorting Genetic Algorithm II

NSGA-II is an Evolutionary Algorithm (EA) for multi-objective optimization. EAs are algorithms which mimic the genetic principle, where only the strongest genes survive [56]. EAs have chromosomes that can be altered, with a set of possibilities for each gene. EAs are often used for solving complex problems such as a Multiobjective Optimization Problem (MOOP) [57]. MOOPs are problems which has multiple objective functions to be minimized or maximized. They may have constraints, which must be satisfied for any feasible solution [58].

NSGA-II has a computational complexity of $\mathcal{O}(MN^2)$, where N is the population size and M is the number of objectives [59]. It uses a non-dominated criteria, where solutions may not be dominated by other solutions, thus being in the Pareto-front. For NSGA-II, parent and child chromosomes are compared from the fronts created by the non-dominated sorting algorithm, where only the best genes survive, i.e. are taken to the next iteration [59].

For EEG analysis, NSGA-II may be used for e.g. channel reduction, method selection and parameter selection [60, 61]. In this work, it is used for finding the best combination of pre-processing, feature extraction and classification method and their parameters, where classification accuracy is to be maximized.

The result of this analysis is a set of solutions, the *pareto front*. To choose a solution, a trade-off between the different objectives is often necessary. Thus, a visualization of the results may help in the choice of a solution.

Chapter 3

Literature Review

In this chapter, earlier related work on MI classification and MI-based BCIs is further explored, in order to give a better understanding of the possibilities for the project. This state-of-the-art review is an extension of the review made in [1]. Next, similar BCIs are presented. Based on this review, some conclusions regarding which methods and approaches to use are drawn.

3.1 Review of State-of-the-art in MI Classification

To get an overview over the performance capabilities of the different methods, a literature review on the state of the art methods was conducted. As there are several studies comparing feature extraction and classification methods for EEG-based BCIs, one of these were used as a basis. The paper used was a meta-study from 2018 [38], where a number of methods for classification and feature extraction were investigated.

For feature extraction, CSP has proved to obtain great results, and has been the golden standard [38]. CSP based algorithms also proved themselves great in BCI Competition III. For a two class MI dataset, dataset IVa, the three highest ranking contenders all used CSP based classifiers [62]. Also, the winners and the four runner ups of the asynchronous MI dataset, dataset I, from BCI Competition IV, all used CSP based algorithms [63]. For dataset 2A in the same competition, all contenders used CSP for feature extraction [63]. This was a multi-class MI dataset, where the winner used a popular Filter Bank Common Spatial Pattern (FBCSP) approach, presented in [34].

In addition to CSP, Lotte et. al. emphasizes Riemannian gemoetry-based classifiers as promising, and proposes them as the new state-of-the-art for MI-based BCI problems [38]. For dataset 2A, a Riemannian approach together with Multivariate Empirical Mode Decomposition (MEMD) is presented in [64], which reportedly scored higher than the competition winner.

For classification, SVM have been, and still is, one of the most popular methods, according to [38]. The method has a good generalization capability, and can therefore obtain good MI classification accuracy [65]. It was also one of the most popular classification methods in BCI Competition III [62]. In addition, LDA is a popular classification algorithm [38], and was also frequently used in BCI Compe-

tition IV [63].

For deep learning networks, such as CNNs, they seem less effective for classifying EEG signals for BCIs. However, this may be due to limited training data available, as deep learning approaches often require large datasets. However, some papers have reported interesting results regarding CNNs of classification of EEG-recorded MI data. In [66], both shallow and deep CNNs were tested, where the shallow CNN reportedly scored higher than the aforementioned Riemannian method for dataset 2A of BCI competition IV. Additionally, Zhao et. al. claimed to achieve an average accuracy of 81.1% with their deeper CNN on the same dataset [67]. According to [38], shallow CNNs are labeled as more promising.

As mentioned, both authors had a specialization project prior to this thesis, where different methods were tested. In [1], a DWT-based feature extraction approach was used with different classifiers, one of which was SVM. Similarly, an Empirical Mode Decomposition (EMD)-based feature extraction with SVM was tested in [2]. The DWT-based classifier scored higher when training and testing on all subjects, and an average higher for the subjects individually. In another study, DWT and EMD were compared for the same dataset, where DWT outperformed EMD vastly [68]. A 1D-CNN was also tested on the empirical wavelet decomposed signal in [2], which scored similar to the EMD SVM classifier. In [1], some Riemannian geometry-based classifiers were tested. With 117-118 electrodes, these methods achieved similar or higher performance than DWT-based methods in two datasets. When using a subset of 9 electrodes, however, the Riemannian geometry-based methods performed worse.

3.1.1 Summary and Methods Chosen

With the information on classification models gained from this literature review, some classifiers are chosen for analysis of self-recorded data. As CSP has been proven to be successful in plenty of studies on MI-based classification, it is implemented and used. Additionally, the newer found success of Riemannian Geometry based classifiers are very interesting, and therefore also implemented in different pipelines. To improve these methods from the work in [1], additional pipelines were tested, and regularization in the Covariance estimates was added. As it was reported successful, a shallow CNN based on the architecture in [66] is implemented. From the authors' earlier work and other studies, DWT has proved better than EMD, and DWT is therefore also used further. In the DWT, the wavelet must be chosen. In [1], several different wavelets were tested. Based on that work, only the wavelet biorthogonal 2.8. is used in this work, as it gave a generally good performance across datasets.

Regarding Riemannian Geometry-based classifiers, some of the pipelines for feature extraction and classification proposed in the literature are: 1) Filtering the covariance matrices with FGDA and classifying with MDRM [51], 2) Projecting the covariance matrices to the TS and classifying with SVM or LR [52, 53], 3) Filtering the covariance matrices with R-CSP and classifying with LDA [55]. These pipelines are explored in this work. Also, because FGDA has shown good performance on noise rejection [51], it is also included in a version of the 3rd pipeline: 4) Filtering the covariance matrices with FGDA, then performing R-CSP and classifying with LDA.

3.2 Similar Studies of BCIs

There have been several studies on drone control using BCIs. In [69], a drone controlling scheme using right hand, left hand, both hands and no hands (resting-state) was implemented. Here, the right and left hand MI induced turns to the respective sides, while both hands made the drone elevate and no hands made the drone decrease in altitude. The drone also had a constant forward velocity. An accuracy of 86.5% of complicated flight tasks was obtained. In [35], an improved FBCSP was tested for four MI classes and one rest class, which resulted in mean accuracy of 41.8+-11.74% in offline classification. The intention was to use these classes as commands for a drone, but the results were not good enough for online implementation.

The winners of dataset I in BCI competition IV, which is an asynchronous dataset, included the resting-state as a class in an uncued BCI [63]. Similarly, in [70], the resting-state was also included as a class to differentiate between MI task and no control intention in continuous data. In contrast, [71] introduced a probability estimator using the two MI classes of dataset IVb of BCI Competition III, where a separate resting-state class was not used. Instead, the combination of probabilities of MI classes was used to calculate the whether or not the subject was resting.

Other papers have presented combinations of MI and other neuro-paradigms, or only other paradigms for BCI control of drones and similar external devices. In [72], a combination of MI and Steady-State Visual Evoked Potential (SSVEP) was used in order to control a quadcopter. SSVEP come from visual stimulation by different frequencies. Thus, by looking at something which is flickering at a certain frequency, the brain responses can classified and used as a command. Left and right hand MI were used to control left-forward and right-forward control respectively, while SSVEP was used to control the altitude. Also, pure SSVEP BCIs have also been introduced, such as in [73], where drone control was successfully implemented. These types of drone controlling schemes rely on external visual stimuli, and are therefore labeled as reactive BCIs.

For drone swarms, other approaches using i.e. visual imagery of the drone movements has also been tested. In [74], a four class visual imagery BCI was introduced for drone swarm control, where each task represented a command for the drone swarm. However, low classification accuracies were achieved, averaging 36.7% for four classes. In [75], MI, Visual Imagery and Speech imagery was tested with the intend of implementing it for drone swarm control. Substantially good classification results were not achieved for either paradigm, thus not making it feasible to implement for drones.

3.2.1 Summary and Chosen Approach

Although other possibilities have been presented, it seems that the approach of including the resting-state as a class in MI-based BCIs is common. In some cases this mental state is mapped to a command, but we argue that a more natural and intuitive approach is to use it as a no-command class. Both successful and unsuccessful MI drone control studies have been presented, showing it as a difficult task. It can also be difficult to use the MI neuro-paradigm in asynchronous BCIs, which have resulted in most MI-based BCI studies being focused on cue-based BCIs [76]. From our literature

study, it also seems that reactive BCIs, using e.g. SSVEP, are more reliable and more successful in general. However, similar to Müller et. al., we believe that active and asynchronous BCIs are more intuitive and useful in a practical setting [8]. As it is difficult achieving high accuracies for multiple MI classes, this thesis approaches the problem for two MI classes at first, then proceeding to add classes if success is achieved with two classes.

Chapter 4

Data Acquisition and System Design

In this chapter, the different individual parts of the data acquisition and drone controlling system is explained. In the start, the headset and EEG recording equipment is presented. After, the individual steps of the sequence illustrated in fig. 4.1 are explained.

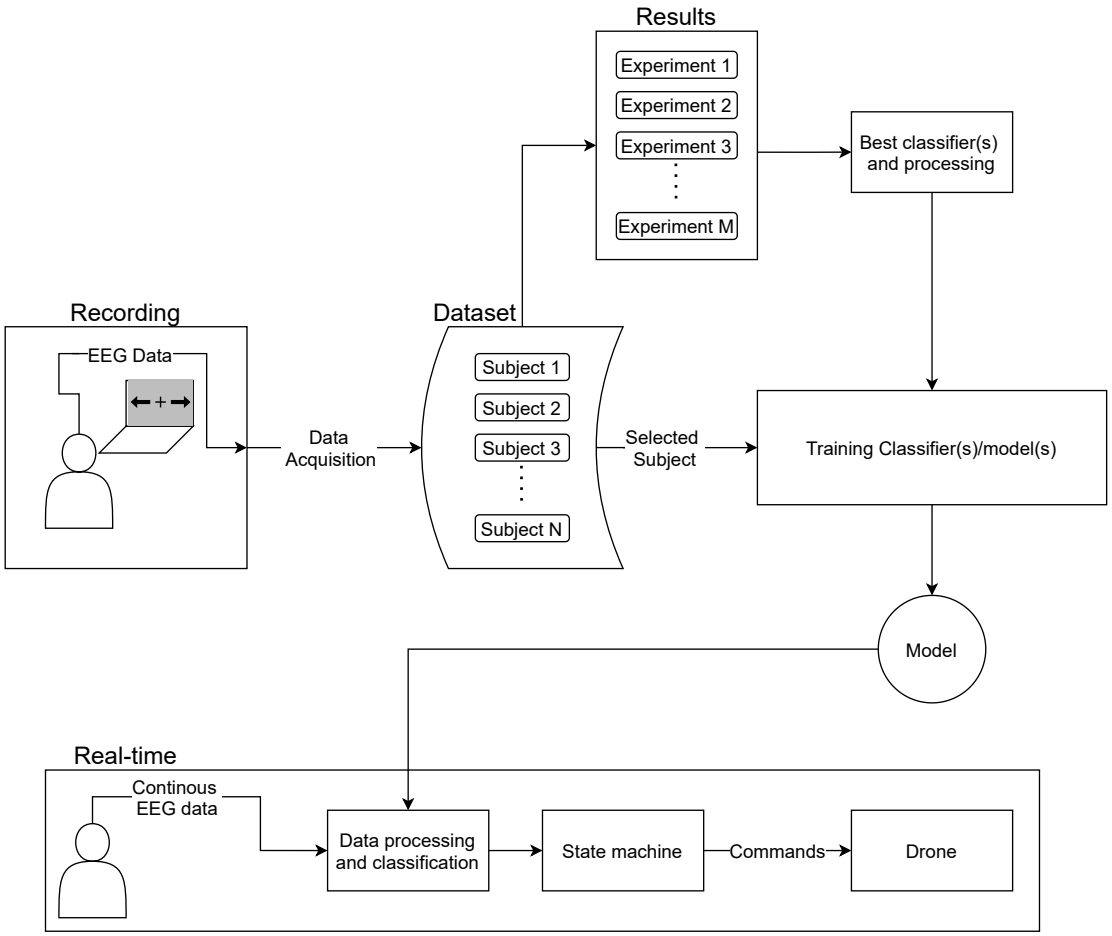


Figure 4.1: Overview of the process from data acquisition, subject and model selection, as well as the real-time system.

These steps include the recording protocol, which is the visuals the subjects are presented with during the data acquisition, presentation of the dataset created and how experimentation and training of classifiers and processing is conducted. After creating a satisfactory model, it is used in real-time to

control a drone. The subsequent steps explain the process of handling continuous EEG data during real-time classification, how the state machines are designed, and how this is translated into commands executed by a drone. Lastly, evaluation of the system is described. These steps are presented in individual sections.

4.1 Equipment and Tools

In this section, the hardware and equipment used is presented.

4.1.1 OpenBCI Headset

To record EEG data, a device called the Ultracortex Mark IV EEG Headset, created by OpenBCI is used. This is a low-cost headset, which has an open source Application Programming Interface (API). For recording with a sample rate of 250 Hz, up to 8 dry comb electrodes with Ag-AgCl coating can be used. In addition, there is one electrode at each earlobe: One reference electrode, and one ground electrode with common-mode noise rejection. The recorded EEG signals are transferred to the computer via Bluetooth, such that the wearer can move freely around. In addition, a velcro strap was attached to the headset, in order to better keep it in place and increase stability of scalp-electrode contact. The headset including the velcro strap is shown in fig. 4.2.

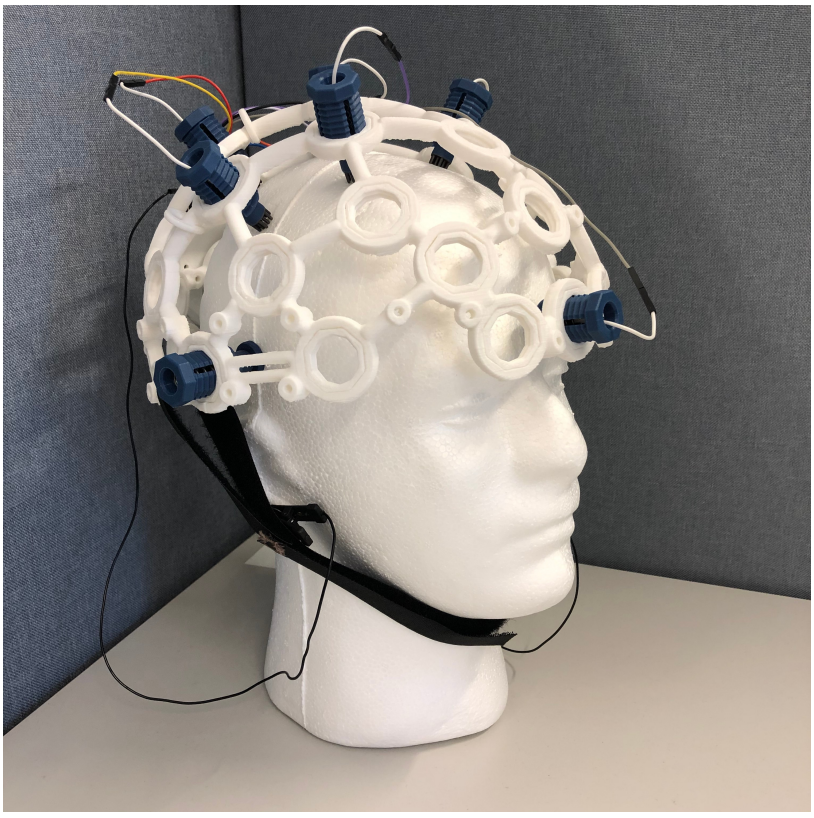


Figure 4.2: Picture of the OpenBCI Ultracortex Mark IV EEG headset including an added velcro strap.

4.1.2 Parrot AR.Drone 2.0

The drone used in the experiments is a quadcopter called Parrot AR.Drone 2.0, and is shown in fig. 4.3. A python API created for the drone, called PS-drone, is used to send commands to the drone [77].

Figure 4.3: Picture of the AR parrot drone 2.0 used for experiments.

4.1.3 The IDUN Computing Cluster

To perform the most computationally heavy tasks, the High Performance Computing system IDUN was used [78]. Specifically, the IDUN cluster was used to run NSGA-II for optimization of methods and parameters for pre-processing, feature extraction and classification.

4.2 Electrode Placement

The OpenBCI Ultracortex headset allows for electrode placement according to the extended international 10-20 system. However, only 35 of the electrode locations are available. See fig. 4.4 for the available electrode locations.

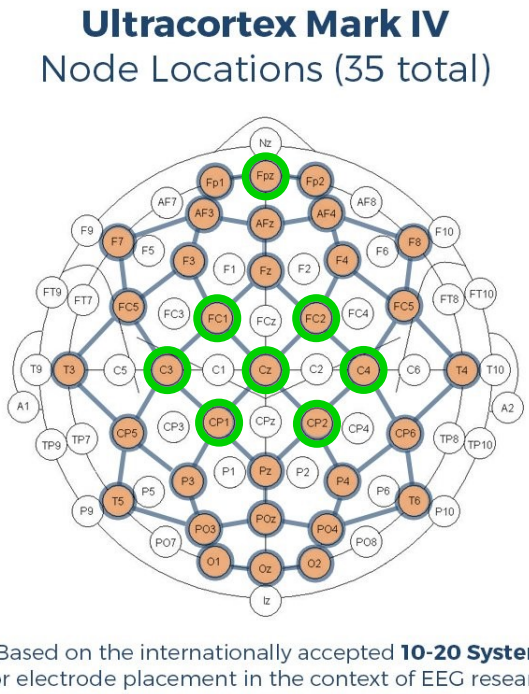


Figure 4.4: Available electrodes in the Ultracortex headset are marked in orange. The electrode placements in use during this thesis are highlighted in green. The figure is adapted from a figure obtained from the OpenBCI webpage [1].

The electrodes used during recording in this work are highlighted in green in fig. 4.4. The electrode positions C3, C4 and Cz are typically associated with right hand, left hand and foot imagery or movement respectively, see section 2.3, and are therefore included. The Ultracortex headset keeps the maximum sample rate of 250 Hz with up to eight electrodes. With more than eight electrodes, the sample rate is reduced to 125 Hz. Because using three or eight electrodes makes no difference for the sample rate in the Ultracortex headset, five more electrodes were added in case they could be useful for analysis. Four electrodes close to C3, Cz and C4 were chosen, namely FC1, FC2, CP1 and CP2. These electrodes were chosen due to their position also being close to the motor cortex and relevant motor control areas. Also, an electrode in the Fpz location was used. This electrode was only added to help in a possible implementation of EOG artifact detection and removal. However, unless it is clearly stated in the experiment, this electrode was not used for analysis.

4.3 Data Acquisition

In this section, we present the recording protocol used for data acquisition. After, information about the subjects are presented. Lastly, the two recorded datasets are described.

4.3.1 Recording Protocol

The recording protocol displays cues to the subject and coordinates the timing of the cues. When designing a protocol for data collection, it was important with a simplistic and easy-to-understand

[1] Ultracortex Mark IV: https://shop.openbci.com/products/ultracortex-mark-iv

protocol, because the majority of the subjects would be new to recording MI EEG data. As explained in **Section 2.5**, visual cues such as arrows, writing or pictures can be used in protocols. As in the examples, the protocol designed for this thesis utilizes arrows to indicate which MI movement to perform. The protocol was also designed to easily be used from two and up to four tasks. This enables the possibility to record using different numbers of tasks by only changing one parameter.

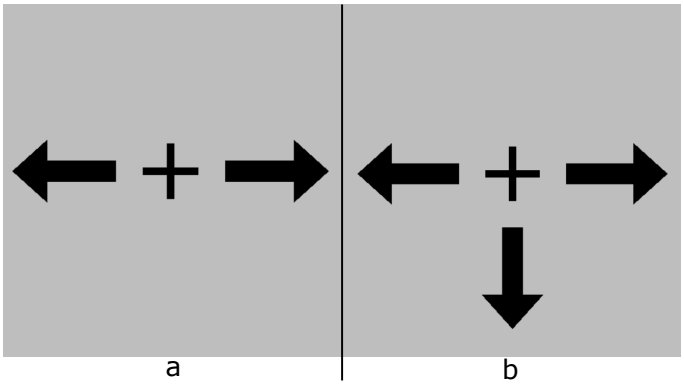


Figure 4.5: The protocol Graphical User Interface for: a) Two classes and b) Three classes

In fig. 4.5, the display of the protocol during the use of different numbers of tasks are shown. In fig. 4.5.a), two arrows are used, indicating the two tasks left hand and right hand MI. These two MI tasks are used for two task recording, as they should be easiest to separate. When adding a third class, the feet are added, as they can also be distinguishable from the other two classes. Foot MI is indicated by the arrow pointed down. There were no feedback given to the user.

In the center between the arrows there is a black cross, which serves as a fixation point. As the subject is focusing at the fixation cross, one of the arrows will turn red, indicating MI performance of the task corresponding to the red arrow. The task indication was set to last for 2.5 seconds, whereas the first 0.5 second is labeled as reaction time. The two following seconds is the part of the data that is used for MI classification. After the task, the arrow turns black again, and the subject is now resting while focusing on the fixation cross. The rest sections is set for 5 seconds, giving the subject enough time to reset and be ready for the next task. This sequence is illustrated in fig. 4.6. To make it compatible with short time windows in real time, short windows for MI was preferred. However, during early system testing, shorter MI periods were tested, but was disregarded, as it was difficult to perform MI in such short time windows. Therefore, the window had to be increased to 2 seconds.

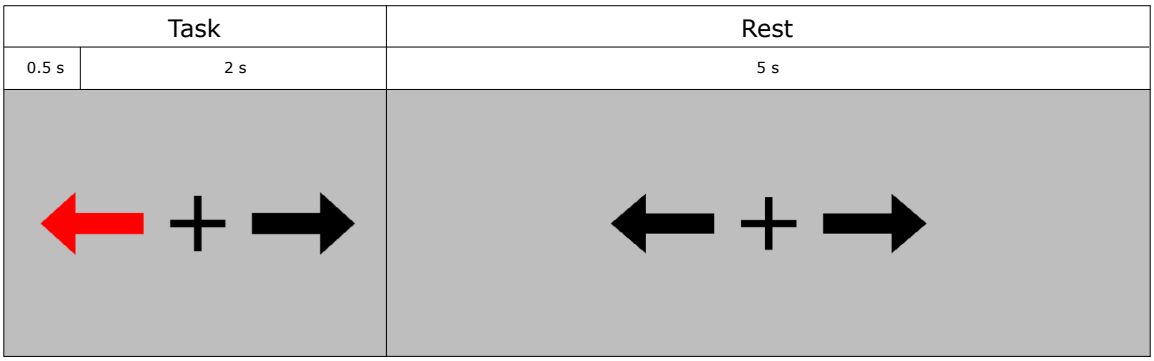


Figure 4.6: Timing scheme for the protocol.

In order to obtain EEG data of good quality, and reduce the chance of variables other than MI in recording, all subjects were instructed to not do any form of body movement, and also keep their eye movement at a minimal. The subjects were told not to move their eyes according to the MI tasks, as this can pollute the data. It was preferred that they kept their eyes on the fixation cross, but they were allowed to change focus point during recording as long as it was done rarely.

For the MI tasks, the subjects were instructed to imagine a hand movement somewhat similar to grasping, but they could choose whichever movement they found the easiest as long as it included moving the hand and fingers. Some suggestions were given, like grasping a ball, grasping in the air, drawing something or playing the piano. It was recommended to use a movement familiar to the subject. For the foot movement, it was recommended to imagine a kick or something similar, such that both the foot and knee would be activated.

During recording, the subject was instructed to sit relaxed and comfortably on a chair in front of the screen displaying the cues. The subject was asked to choose between resting their arms on their lap, or along their body, whichever was more comfortable and less distracting. During the rest stages, the subjects were instructed to avoid thinking about any movements, to reduce chances of contaminated data.

4.3.2 Subjects

A total of 16 subjects were invited to perform MI recordings. All of the subjects are able-bodied individuals in the ages of 23-27 years. Of the 16 subjects, 9 are females and 7 are males. Of the 16, only three had experience with MI before joining the experiments. Underneath, in table 4.1, the sex(F/M), age, laterality(L/R) and prior MI experience(Yes/No) is presented. The data collection and handling was approved by Norwegian Centre for Research Data (NSD).

Table 4.1: Sex, age, laterality and prior MI experience of the 16 subjects.

Subject	Sex	Age	Laterality	Prior MI experience
1	F	23	R	Yes
2	F	25	R	Yes
3	F	23	R	No
4	F	25	R	No
5	M	25	R	No
6	F	24	L	No
7	F	24	R	No
8	M	24	L	No
9	F	23	L	No
10	F	24	R	No
11	M	27	R	No
12	F	23	R	No
13	M	27	R	No
14	M	24	R	No
15	M	23	R	No
16	M	26	R	Yes

4.3.3 Dataset A

Dataset A was the first part of the data being recorded. All 16 subjects participated in four recording sessions, over two days with 1-7 days between. EEG data was recorded using two MI tasks. Each session had 30 trials of each MI task, making one session approximately 8 minutes and 30 seconds. Thus, **dataset A consists of data from 16 subjects executing right hand and left hand imagery movement, with 4 sessions per subject and 60 trials per run. Per subject, this is a total of 34 minutes of EEG data and 240 MI trials.**

The first of the two recording days included the following procedure for each subject: First, the MIQ-3 questionnaire was conducted, giving the subject an introduction to MI [79]. Then, the subject was given verbal guidance on MI and a demonstration of the protocol. After the subject had no more questions about the protocol or performing MI, the subject was given five minutes to practice the tasks. It was recommended to perform the task physically a few times before practicing the mental execution. Next, the Ultracortex EEG headset was fit to the subject's head, and the first run was started. After the first run, the subject was allowed to rest for 2-5 minutes before the second run.

Through the MIQ-3 questionnaire, the subjects were introduced to the three ways to perform MI (internal visual, external visual, kinestethic), and obtained a feeling of what they found easier. The results of the MIQ-3 questionnaire are not presented in this work, as it was only used for introducing the subjects to MI and the results were not analysed.

The procedure for the second of the two recording days was the same, except that the MIQ-3 questionnaire was not conducted and the verbal guidance and protocol demonstration was not repeated. The subject was however still allowed to ask questions about the procedure and still given five minutes practice.

4.3.4 Dataset B

The second part of the data was recorded with subject 1, 2, 10, 15 and 16. These subjects were chosen because of better results for Dataset A. In this part of the dataset, foot imagery was added, which means three MI tasks were used - left hand, right hand and foot. The 6 subjects participated in 2 recording sessions, recorded in one day. In each session, there were 20 trials of each MI task, to keep the same total time span of each run as in Dataset A. This resulted in a total of 40 trials for each MI task, with a total of 120 instances.

4.4 Epoching the Data

In this section, the ephoching of both offline and online experiments are explained.

4.4.1 Offline Epoching

The data from each session is saved as continuous data. To create a classifier, only selected parts of the continuous data is used. From each trial, the last two seconds of the 2.5 seconds task indication is used, as the first 0.5 seconds are labeled as reaction time. In addition, the middle 2 seconds of the 5 seconds long rest period is used for classification, to give the possibility of hovering/resting for the drone. Only the middle part of rest period is used, as it was considered as less likely to have overlap with MI performance. By extracting this part of the data, it reduces the chances of training on data containing MI from the task periods, in cases of prolonged execution of MI from the subject. Also, it reduces the chances of the subject preparing for the next task. The data extraction discussed is illustrated in fig. 4.7.

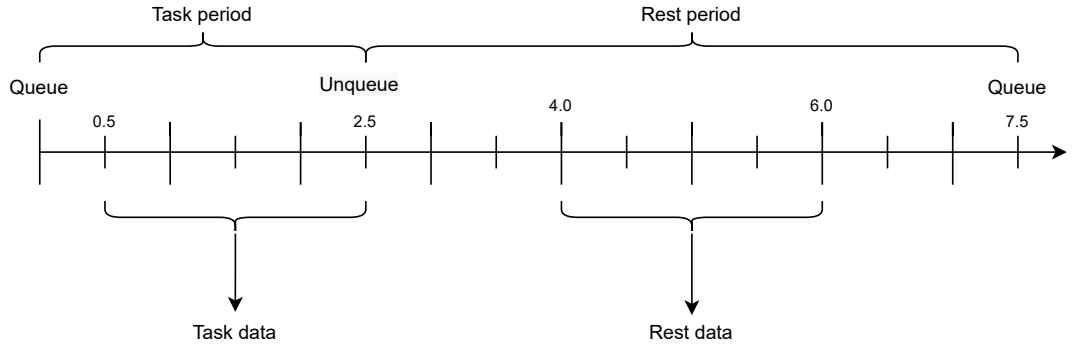


Figure 4.7: Visualization of offline epoching of the MI tasks and resting-state data.

4.4.2 Online Epoching

As the EEG recording equipment will produce a continuous stream of EEG data, it must be segmented such that the online data adheres to the same structure as the training data. Therefore, the stream of data is continuously epoched into trials of 500 samples, each corresponding to 2 seconds. This simple solution is illustrated in fig. 4.8.This will give a new classifier output roughly every 2 seconds.

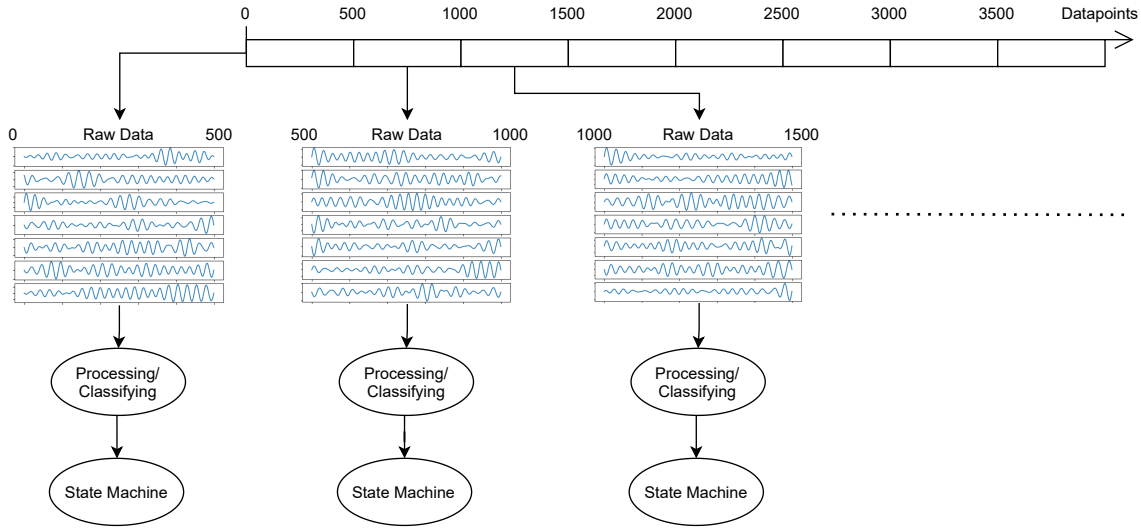


Figure 4.8: Visualization of the online epoching

4.5 From Raw Epochs to Classifier Output

In this section, the general pipeline for this thesis is explained. In fig. 4.9, the pipeline is illustrated. The raw data is first notch filtered at 50 Hz and high pass filtered at 0.1 Hz. After, re-referencing can be executed, but it is optional. Either way, a band pass filter or decomposition by DWT is introduced. Then, features are extracted from the signal, which is then used for classification. However, there is one exception, which is during classification with CNN. Then, the feature extraction and classification is in the same step, as the neural network learns features by itself.

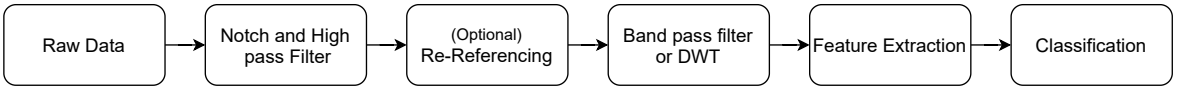


Figure 4.9: Generalization of the data processing and classification pipeline used in this thesis

The specific methods to use in each step in order to obtain the highest classification accuracy are determined in the experiments described in the next chapter.

4.6 State Machine

A state machine retains some information about the the current state of the drone, which is used in combination with classification results to send the correct command to the drone in real-time. As control with both two and three MI tasks is implemented, two different state machines are presented.

Both are explained in the subsequent subsections.

As MI classification can induce a high False Positive Rate (FPR) in the resting-state, including a layer between the classification output and the control output is advantageous to filter false positives. Also, in uncued BCIs, one execution of MI may overlap partly with multiple time windows. To prevent multiple commands for what was intended to be one MI execution, a simple safety mechanism is added. By having a *blocking state*, which requires the resting-state to be the previous mental state detected as a demand for executing a new drone command, the number of false positives can be reduced.

4.6.1 Drone Operation with Two MI Tasks

For the two task operation, only two commands were available. To retain the possibility of moving in one plane, the horizontal plane, rather than only in a line, one MI task is set to correspond to forward movement, while the other task corresponds to a clockwise rotational movement of approximately 90 degrees.

The state-machine is implemented with automatic lift-off at the initiation of the system. The state-machine is illustrated in fig. 4.10. The two MI tasks are as mentioned corresponding to each movement, while a third mental state, resting-state, gives no control output and results in the drone only hovering in place.

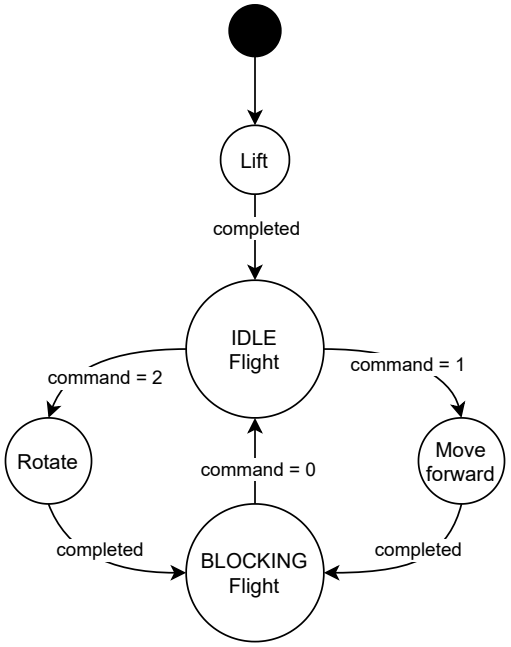


Figure 4.10: State machine for two-dimensional drone operation in the horizontal plane, using two MI tasks. The command 0 correspond to the resting-state, 1 to left hand MI, and 2 to right hand MI.

4.6.2 Drone Operation with Three MI Tasks

The state-machine for three classes is more complex, as there are more possible actions. To take the maximum advantage of the additional class, i.e. foot MI, it is implemented as a *switch* between

vertical and horizontal movement. The horizontal state uses the same movements as the two MI class system, where left MI hand corresponds to forward movement and right hand MI corresponds to a rotation movement. However, when activating the switching by foot MI, the drone will now move vertically. Now, left hand MI corresponds to ascending and right hand MI corresponds to descending. If the added class only corresponds to a new movement, one more possible movement for the drone is obtained. If such a switch is used, two more control options are obtained. The switch mechanism is first illustrated in a simplified version of the state machine in fig. 4.11.

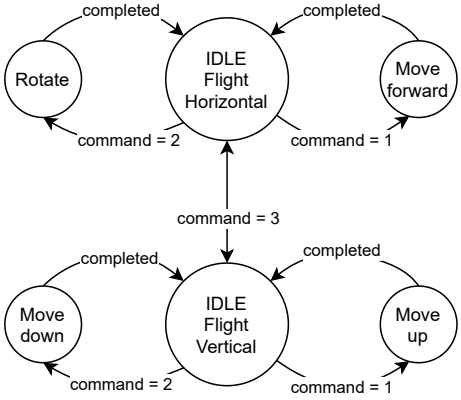


Figure 4.11: Simplified state machine for three MI tasks during the flight state. The command 1 correspond to left hand MI, 2 to right hand MI and 3 to foot MI.

The full state-machine for movement in the three-dimensional space is illustrated in fig. 4.12. This state-machine starts off in the ground vertical state. Then, the only available action for the drone is to take off from the ground. As class 1, left hand MI, corresponds to ascending, it will be used to take off. All other classes makes the drone stay in place. When the drone has executed lift-off, it will be in a flight state. During flight state, both the vertical and horizontal state can be used. If foot MI is detected, while in flight mode, it will change between vertical and horizontal states. A counting mechanism is used to track the approximate elevation of the drone and know whether it is close to the ground and should land upon a *move down* command.

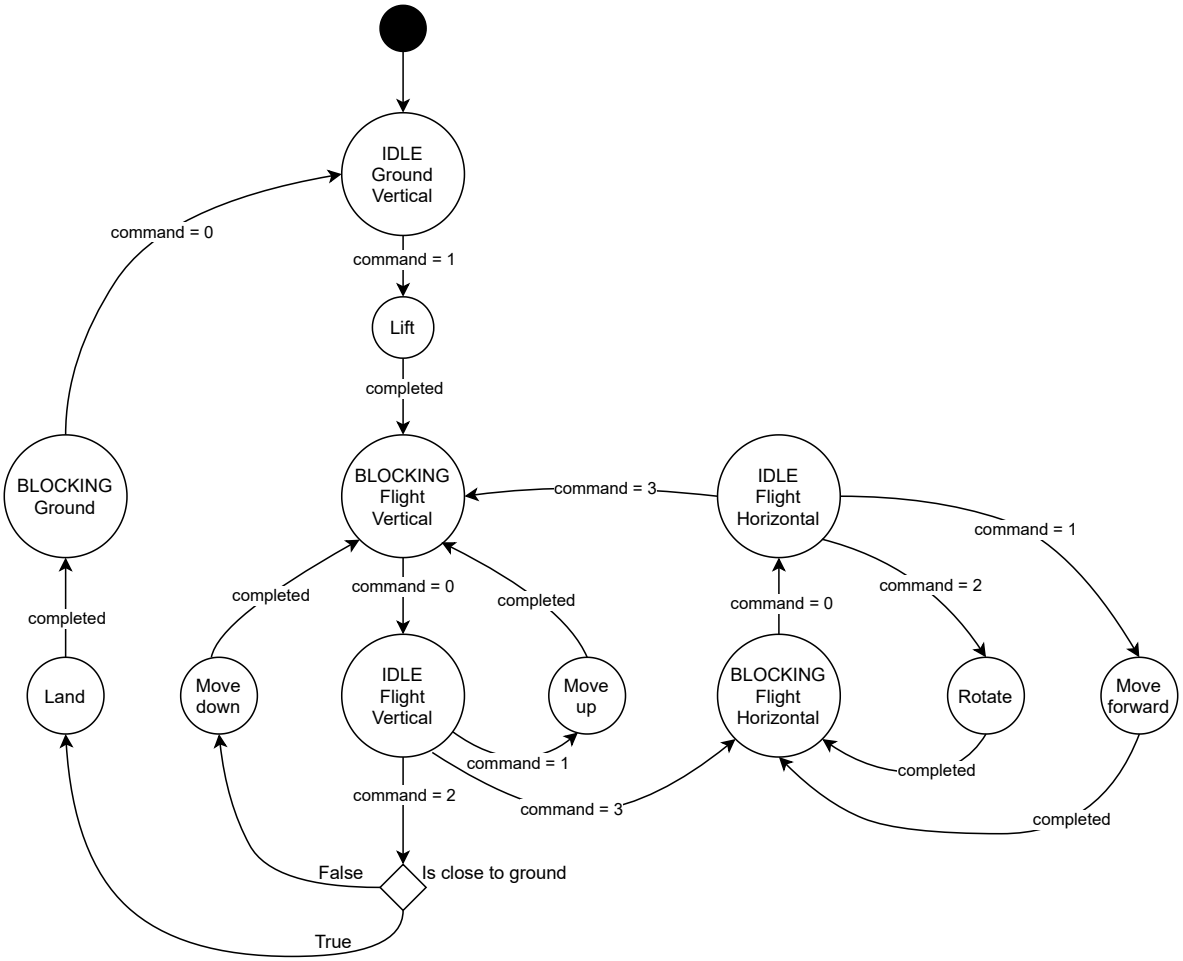


Figure 4.12: Full state machine for three-dimensional drone operation with three MI classes. The command 0 corresponds to resting-state, 1 to left hand MI, 2 to right hand MI and 3 to foot MI.

4.7 Metrics for Evaluation

In order to evaluate the algorithms' ability to classify the EEG epochs and to evaluate the final BCI, it is necessary to define some metrics.

4.7.1 Metrics in Offline Classification

To calculate the *accuracy* in classification, it is necessary to have one set of data for training the algorithm, and one set of data for testing. The resulting accuracy is dependent on which data is used in which set. Therefore, to evaluate algorithms on a single dataset without a well-defined split, *k-fold cross-validation* can be used. In offline classification, accuracy or classification accuracy strictly refers to the 10-fold cross-validation accuracy in this work, unless a training and test set is clearly specified.

Moreover, the confusion matrix is often used for evaluating the results, as an addition to the classification accuracy. In these cases, the *normalized* confusion matrix is always used. Here, each row sums to one, and the matrix shows the *rates* rather than number of instances in each cell.

4.7.2 Evaluating the BCI

For evaluating the BCI during online classification, the True Positive Rate (TPR) and FPR is used. In self-paced BCIs, these metrics can be defined as following [22]:

TPR: The probability of detecting the correct active class (MI task) when MI is performed.
FPR: The probability of falsely detecting an active class when the user has no control intentions (resting-state).

These are the definitions used in this work. Furthermore, several other metrics exist. In the the last chapter, the system is also evaluated in terms of *throughput* and *response time*. Throughput is the number of possible commands to send per time, while response time is the time from the user's intended command to the actuation.

Chapter 5

Results

In this chapter, the results from different experiments are described. The first experiment is an evaluation of the performance of the subjects. Next, data is visualized to show examples of the appearance and absence of ERD/ERS patterns in high- and low-performing subjects. After, an initial analysis of performance with three MI tasks is performed. Then, methods and parameters for pre-processing, feature extraction and classification are optimized with NSGA-II and analyzed. In the same experiment, a hierachical structure of the classifier is compared to the standard flat structure. For the two best-performing subjects, subject-specific methods are selected. After, a cross-session test and a simulation of online classification is conducted, in order to assure viability of online classification, before the online experiments with the drone are presented. Lastly, an analysis of the possible effects artifacts has on the results is carried out.

5.1 Evaluation of Subject Performance

To get an overview of how each subject has performed during the two task MI EEG recording sessions, the data from each subject is evaluated with an initial test. To make the evaluation of the subjects independent of the feature extraction and classification methods, the evaluation is carried out with multiple methods. However, a large amount of possible combinations for feature extraction and classifiers were explored in this thesis, and evaluating each subject with each of these combinations would be too extensive and time consuming. Therefore, only 5 different combinations of feature extractions and/or classifiers are used. This provides an indication of which subjects are performing best, whose methods can later be optimized. In addition, the best subjects are invited for additional recording with three classes.

For the initial test, these five classifiers are used:

- CSP with SVM
- CSP with LDA
- DWT with SVM
- CNN
- Riemannian geometry-based method: Covariance estimate, FGDA and MDRM

For all methods, a notch filter of 50 Hz and a high pass filter of 0.1 Hz were used. For CSP-based methods, CNN and Riemann, an additional bandpass filter of 8-12 Hz was added. For the DWT, two of the sub-bands, D3 and D4, from the four levels of decomposition were used. The achieved accuracies are presented in individual tables for each subject. Here, each session has been scored individually. Additionally, all the sessions accumulated are also scored, and noted as "All" in the tables.

For this section, the two MI classes are classified, and the resting-state is excluded. This is done to easier assess the ability of inducing ERD/ERS during MI in individual subjects. Thus, an accuracy of 50% indicates randomness.

Performance of Subject 1

Subject 1 had prior to the recordings tried EEG-recorded MI, and was therefore familiar with the process. From the results of the initial tests, presented in table 5.1, the subject appeared to have mastered the MI in a moderate way. Both CSP methods and the Riemannian method scored around 65-70%. CNN scored low for the single runs, which may be because of the low amount of data. CNN scored 60% when using all the data, also indicating a performance better than random. However, DWT did not seem to get good results.

Table 5.1: Accuracy in classification of left and right hand MI for subject 1 using different classification methods.

Method/Session	**1**	**2**	**3**	**4**	**All**
CSP + SVM	60.00 %	75.00 %	61.67 %	73.33 %	65.42 %
CSP + LDA	58.33 %	71.67 %	61.67 %	68.33 %	65.83 %
DWT + SVM	53.33 %	46.67 %	60.00 %	60.00 %	45.00 %
CNN	50.00 %	55.00 %	45.00 %	55.00 %	60.00 %
RIEMANN	63.33 %	75.00 %	60.00 %	65.00 %	70.42 %

Performance of Subject 2

Similarly to the first subject, subject 2 also had experience with EEG-recorded MI. The results from the initial tests indicated a good MI performance. For both CSP methods and the riemannian method, an accuracy above 80% was achieved. Both DWT and CNN performed somewhat lower, but still resulted in a respectable 71-74%. It is also noted that for the second run in the first session, DWT resulted in a surprisingly low accuracy of 36.67%.

Table 5.2: Accuracy in classification of left and right hand MI for subject 2 using different classification methods.

Method/Session	1	2	3	4	All
CSP + SVM	88.33 %	80.00 %	78.33 %	75.00 %	85.83 %
CSP + LDA	88.33 %	78.33 %	75.00 %	75.00 %	83.75 %
DWT + SVM	83.33 %	36.67 %	78.33 %	78.33 %	71.67 %
CNN	66.67 %	58.33 %	70.00 %	66.67 %	74.17 %
RIEMANN	90.00 %	80.00 %	80.00 %	83.33 %	80.83 %

Performance of Subject 3

In contrary to the first two subjects, the initial tests indicate a random performance, as all methods scored an accuracy between 52% and 55% when testing on the accumulated data. Therefore, it is believed that no ERD/ERS was evoked during the MI tasks, making it difficult to distinguish the two classes.

Table 5.3: Accuracy in classification of left and right hand MI for subject 3 using different classification methods.

Method/Session	1	2	3	4	All
CSP + SVM	50.00 %	53.33 %	53.33 %	55.00 %	54.17 %
CSP + LDA	46.67 %	51.67 %	55.00 %	56.67 %	54.58 %
DWT + SVM	36.67 %	55.00 %	60.00 %	65.00 %	54.17 %
CNN	50.00 %	51.67 %	55.00 %	35.00 %	52.83 %
RIEMANN	70.00 %	58.33 %	55.00 %	56.67 %	55.42 %

Performance of Subject 4

The initial test of subject 4 indicates an ability of performing MI, as classification accuracies on the accumulated data was between 60% and 65% for all classifiers. It seems that the first session was performed well, but with a general decreasing trend for the runs. If this is due to some random factors in the signal, a source of error or less prominent ERD/ERS is hard to tell.

Table 5.4: Accuracy in classification of left and right hand MI for subject 4 using different classification methods.

Method/Session	1	2	3	4	All
CSP + SVM	71.67 %	61.67 %	58.33 %	55.00 %	63.33 %
CSP + LDA	70.00 %	56.67 %	60.00 %	53.33 %	64.17 %
DWT + SVM	61.67 %	51.67 %	50.00 %	40.00 %	62.50 %
CNN	58.33 %	43.33 %	40.00 %	48.33 %	60.00 %
RIEMANN	71.67 %	61.67 %	65.00 %	50.00 %	63.75 %

Performance of Subject 5

For subject 5, both better and worse than 50% was acquired for individual sessions. For the accumulated data, both CSP methods resulted in barely above 60% accuracy. For this subject, the results for individual and combined runs were not convincingly good, although some patterns seems to have been discovered.

Table 5.5: Accuracy in classification of left and right hand MI for subject 5 using different classification methods.

Method/Session	1	2	3	4	All
CSP + SVM	56.67 %	63.33 %	48.33 %	51.67 %	61.25 %
CSP + LDA	58.33 %	61.67 %	43.33 %	56.67 %	60.83 %
DWT + SVM	55.00 %	60.00 %	58.33 %	56.67 %	56.67 %
CNN	48.33 %	50.00 %	50.00 %	61.67 %	52.91 %
RIEMANN	56.67 %	66.67 %	43.33 %	61.67 %	58.75 %

Performance of Subject 6

For subject 6, the accuracies acquired from all methods are promising. Here, CNN and the Riemannian method scored best, giving 70.83% and 75.42% accuracy respectively for all data combined. However, there is an interesting trend in the individual sessions. In general for the methods, the subject has an increasing accuracy for each recording. This can indicate a learning curve for the subjects ability to perform MI.

Table 5.6: Accuracy in classification of left and right hand MI for subject 6 using different classification methods.

Method/Session	1	2	3	4	All
CSP + SVM	50.00 %	61.67 %	75.00 %	78.33 %	62.50 %
CSP + LDA	58.33 %	66.67 %	70.00 %	75.00 %	63.33 %
DWT + SVM	46.67 %	53.33 %	66.67 %	56.67 %	60.83 %
CNN	61.67 %	65.00 %	68.33 %	75.00 %	70.83 %
RIEMANN	56.67 %	65.00 %	76.67 %	76.67 %	75.42 %

Performance of Subject 7

For subject 7, tests on both individual sessions and accumulated data are indicating randomness. The results, which are presented under in table 5.7, gives no reason to believe that the MI performed evoked the desired responses in the brain.

Table 5.7: Accuracy in classification of left and right hand MI for subject 7 using different classification methods.

Method/Session	1	2	3	4	All
CSP + SVM	48.33 %	40.00 %	48.33 %	48.33 %	53.75 %
CSP + LDA	50.00 %	41.67 %	58.33 %	51.67 %	53.33 %
DWT + SVM	50.00 %	38.33 %	41.67 %	50.00 %	40.00 %
CNN	60.00 %	48.33 %	38.33 %	56.67 %	46.67 %
RIEMANN	51.67 %	31.67 %	48.33 %	46.67 %	50.00 %

Performance of Subject 8

Subject 8 had interesting results in the individual sessions. Both session 1 and 3 achieved good results for both CSP-based methods and Riemann. However, session 2 and 4 seems to have close to random results. As session 1 and 2 was recorded the same day, and session 3 and 4 was recorded in the same day, it might be related to fatigue. The two low performing sessions likely made the accumulated data score lower.

Table 5.8: Accuracy in classification of left and right hand MI for subject 8 using different classification methods.

Method/Session	1	2	3	4	All
CSP + SVM	68.33 %	51.67 %	71.67 %	58.83 %	61.67 %
CSP + LDA	71.67 %	56.67 %	70.00 %	51.67 %	59.58 %
DWT + SVM	43.33 %	50.00 %	61.67 %	48.33 %	54.16 %
CNN	55.00 %	48.33 %	63.33 %	56.67 %	61.67 %
RIEMANN	63.33 %	50.00 %	70.00 %	48.33 %	62.50 %

Performance of Subject 9

The initial tests indicate a bad performance, as seen in table 5.9. For the second run in the second session, CSP with LDA even scores as low as 33.33%. Accuracies of close to 50% are achieved for the accumulated data, indicating randomness.

Table 5.9: Accuracy in classification of left and right hand MI for subject 9 using different classification methods.

Method/Session	1	2	3	4	All
CSP + SVM	56.67 %	48.33 %	53.33 %	35.00 %	52.50 %
CSP + LDA	53.33 %	50.00 %	58.33 %	33.33 %	55.00 %
DWT + SVM	48.33 %	63.33 %	45.00 %	50.00 %	50.83 %
CNN	53.33 %	43.33 %	38.33 %	46.67 %	50.42 %
RIEMANN	48.33 %	46.67 %	53.33 %	43.33 %	58.33 %

Performance of Subject 10

As seen in table 5.10, the classification accuracies for individual sessions and all data together obtained a decent score, with up to 80% accuracy for individual runs, and just below 70% accuracy for all runs together. This is a good performance, indicating the correct brain patterns are evoked.

Table 5.10: Accuracy in classification of left and right hand MI for subject 10 using different classification methods.

Method/Session	1	2	3	4	All
CSP + SVM	71.67 %	80.00 %	73.33 %	65.00 %	66.67 %
CSP + LDA	68.33 %	76.67 %	68.33 %	63.33 %	65.00 %
DWT + SVM	61.67 %	53.33 %	61.67 %	55.00 %	57.08 %
CNN	53.33 %	56.67 %	51.67 %	58.33 %	62.92 %
RIEMANN	53.33 %	76.67 %	71.67 %	61.67 %	69.58 %

Performance of Subject 11

The results from the initial tests, shown in table 5.11, suggests that MI-related ERD/ERS patterns were not clear, as none of the algorithm managed to perform well in classification. Individual and accumulated data are all indicating randomness.

Table 5.11: Accuracy in classification of left and right hand MI for subject 11 using different classification methods.

Method/Session	1	2	3	4	All
CSP + SVM	43.33 %	61.67 %	53.33 %	53.33 %	51.25 %
CSP + LDA	41.67 %	60.00 %	53.33 %	53.33 %	52.92 %
DWT + SVM	51.67 %	51.67 %	61.67 %	55.00 %	52.92 %
CNN	51.67 %	45.00 %	50.00 %	48.33 %	50.83 %
RIEMANN	40.00 %	55.00 %	48.33 %	58.33 %	57.08 %

Performance of Subject 12

With generally low classification results, it seems that the brain responses during MI were not convincing for subject 12. While having some individual runs gaining accuracies above to 60%, other runs had down to 40% with the same method. This could be due to the algorithms not finding any reliable patterns, resulting in classification accuracies of $50 \pm 10\%$, and no accuracies above 60% for the accumulated data.

Table 5.12: Accuracy in classification of left and right hand MI for subject 12 using different classification methods.

Method/Session	1	2	3	4	All
CSP + SVM	53.33 %	45.00 %	55.00 %	60.00 %	59.17 %
CSP + LDA	55.00 %	40.00 %	56.67 %	63.33 %	58.33 %
DWT + SVM	51.67 %	58.33 %	41.67 %	63.33 %	55.42 %
CNN	36.67 %	40.00 %	41.67 %	60.00 %	49.58 %
RIEMANN	48.33 %	41.67 %	60.00 %	61.67 %	53.33 %

Performance of Subject 13

The results of the initial tests seem random. This can be seen in table 5.13. This suggests that the attempted MI did not invoke brain responses in the manner it should.

Table 5.13: Accuracy in classification of left and right hand MI for subject 13 using different classification methods.

Method/Session	1	2	3	4	All
CSP + SVM	45.00 %	53.33 %	60.00 %	45.00 %	48.75 %
CSP + LDA	50.00 %	48.33 %	50.00 %	50.00 %	49.58 %
DWT + SVM	35.00 %	55.00 %	41.67 %	56.67 %	50.42 %
CNN	50.00 %	68.33 %	50.00 %	53.33 %	50.42 %
RIEMANN	46.67 %	58.33 %	51.67 %	48.33 %	54.17 %

Performance of Subject 14

As the accuracies obtained with different algorithms did not exceed 60%, as shown in table 5.14, it suggests that the subject did not evoke the right responses in the brain when trying to perform the MI for left and right hand.

Table 5.14: Accuracy in classification of left and right hand MI for subject 14 using different classification methods.

Method/Session	1	2	3	4	All
CSP + SVM	60.00 %	41.67 %	46.67 %	58.33 %	57.08 %
CSP + LDA	58.33 %	38.33 %	55.00 %	55.00 %	58.33 %
DWT + SVM	41.67 %	41.67 %	58.33 %	53.33 %	55.83 %
CNN	51.67 %	51.67 %	53.33 %	41.67 %	52.50 %
RIEMANN	53.33 %	33.33 %	58.33 %	55.00 %	57.92 %

Performance of Subject 15

For the initial tests of the data, the performance is overall great over sessions and classifiers, as seen in table 5.15. The subject seemed to improve during the second session, where both runs scored 90% and higher for three classifiers. This indicates well defined brain responses during MI, and easy distinguishable brain responses.

Table 5.15: Accuracy in classification of left and right hand MI for subject 15 using different classification methods.

Method/Session	1	2	3	4	All
CSP + SVM	73.33 %	71.67 %	93.33 %	90.00 %	82.08 %
CSP + LDA	75.00 %	73.33 %	95.00 %	90.00 %	82.50 %
DWT + SVM	75.00 %	75.00 %	73.33 %	73.33 %	75.42 %
CNN	66.67 %	68.33 %	73.33 %	68.33 %	77.92 %
RIEMANN	70.00 %	75.00 %	93.33 %	91.67 %	82.50 %

Performance of Subject 16

For the initial classification the subject performed decent, as shown in table 5.16. Only DWT obtained an accuracy lower than 60% for the accumulated data. In addition, this subject was the only one who generated highest classification accuracy with CNN.

Table 5.16: Accuracy in classification of left and right hand MI for subject 16 using different classification methods.

Method/Session	1	2	3	4	All
CSP + SVM	70.00 %	70.00 %	66.67 %	63.33 %	62.50 %
CSP + LDA	63.33 %	71.67 %	61.67 %	63.33 %	64.58 %
DWT + SVM	53.33 %	66.67 %	63.33 %	55.00 %	56.50 %
CNN	51.67 %	70.00 %	63.33 %	51.67 %	70.00 %
RIEMANN	66.67 %	68.33 %	56.67 %	61.67 %	67.08 %

5.1.1 Choosing Subjects for Further Analysis

From the initial tests, a large difference in classification accuracies were found for the different subjects. For subjects not evoking the expected MI-related brain responses, it is not possible to create MI based classification algorithms and implement it in a drone controlling scheme, as most commands will appear randomly. Including all subjects may limit the possibility of gaining an optimized classification scheme, due to random non-MI signals. Thus, only a few selected subjects are focused on when trying to optimize the classification algorithms further.

As mentioned, a classification score of 50% is statistically the same as classifying by chance, and is considered the baseline score. However, as can be seen from the score of subject 9, session 2.2. with classification by CSP and SVM, it is possible to achieve scores as low as 35%, 15% below the chance level. This could in some cases correspond to the pattern being detected in the training data being somewhat shifted in the test data, this is likely not the case since 10-fold cross-validation is used. Also, such results could occur due to unbalanced datasets, but these datasets are balanced. Thus, we assume that this result may have been achieved by chance. It is not analyzed further why these

below-random accuracies occurred. Therefore, in the evaluation of the performance of the subjects one must consider that it could also be possible to achieve a score of around 15% above the chance level with CSP and SVM, i.e. around 65%, by chance, in this experiment.

Two subjects, namely subject 2 and 15, performed well above every other subjects, as both high accuracies and high consistency in performance were achieved. When considering the results of CSP with SVM, subjects 1, 6, 10 and 16 also acquired decent results and had a quite consistent performance. Therefore, these six subjects were invited for recording with three MI classes, in order to see if high accuracies could be obtained. These six subjects were also subject to further analysis.

The data from the remaining 10 subjects is excluded in further experiments, except from data of subject 7 being used for visualization in section 5.2. It was assumed that the 10 remaining subjects are either BCI illiterates, or would need training before reaching a satisfying performance. With limited time available, training the subjects was not possible.

5.2 Visualizing the ERD/ERS Pattern

In order to help understand the differences in classification accuracy, a visual analysis is conducted. To do so, two subjects are used, one well-performing subject and one bad-performing. The two subjects chosen for this analysis are subject 2 and subject 7.

First, an inspection of some notch and band-passed signals are evaluated, to see if an ERS/ERD is apparent. After, average FFTs over multiple tasks before and after CSP filtering are analysed. Lastly, a quick look into the energy features extracted from the discrete wavelet transformed signal is given.

5.2.1 Mu ERD/ERS in Time Domain

Finding visible ERD/ERS in the recorded data in time domain proved itself quite difficult, even in high-performing subjects. Nevertheless, two examples are shown in fig. 5.1 and fig. 5.2. These were the examples closest to ERD/ERS found for both subjects.

For subject 2, it seems like ERD is induced during the task in channel C3, which is the channel we would expect ERD. It also seems like ERS is happening during the last part of the task in channel C4, where we also would expect ERS for right hand imaginary movement.

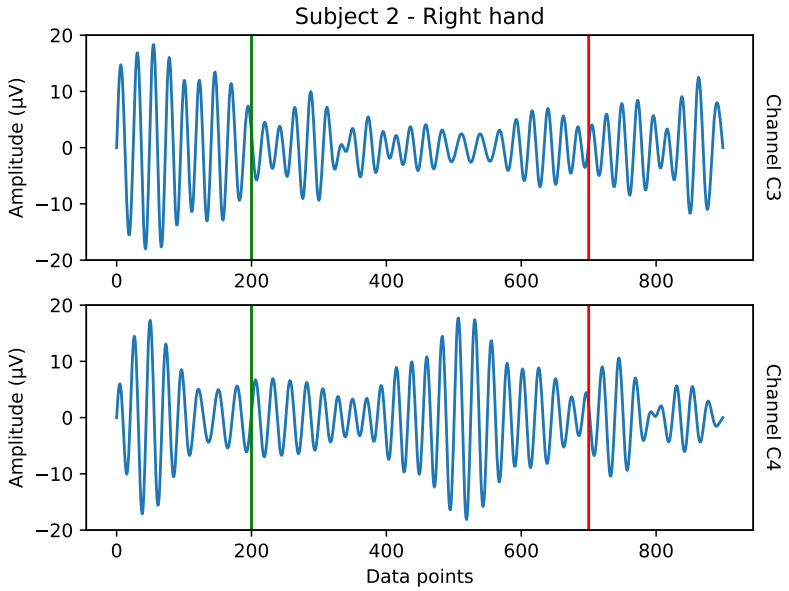


Figure 5.1: Possible ERD/ERS in subject 2 in channels C3 and C4 during right hand MI. The green line indicates the start of a task, and the red line indicates the end of the task.

For subject 7, the possible ERD in channel C3 is less obvious, as it happens for a brief moment, just before 600 points. In the other channel, C4, the subject has no changes, which may be due to the position of this electrode corresponding to the position of a non-attended body-part in the motor cortex.

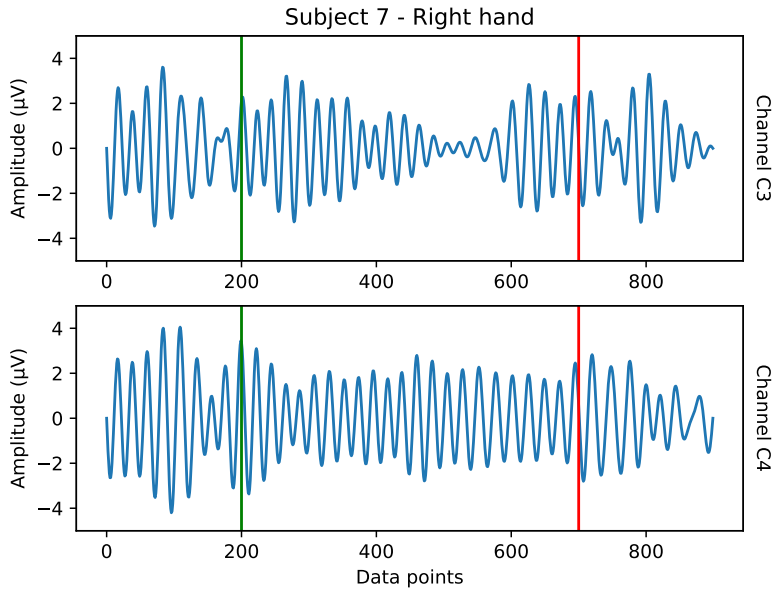


Figure 5.2: Possible ERD/ERS in subject 7. The green line indicates the start of a task, and the red line indicates the end of the task.

In both figures, the possible ERD/ERS patterns can be hard to find. To find a good visualisation was hard, and is therefore not reliable, as it was not appearing on a general basis in the signals.

5.2.2 FFT Before and After CSP Filtering

As visuals of ERD/ERS can be hard to find in time domain, it can be interesting to look at the power of different frequencies, thus making the FFT of the signal interesting. For this, the average FFT values for each class is plotted. In addition to doing this for the notch and band pass filtered signal, it is also plotted for the CSP filtered signal. Thus, how the CSP is able to enhance pattern is also presented visually.

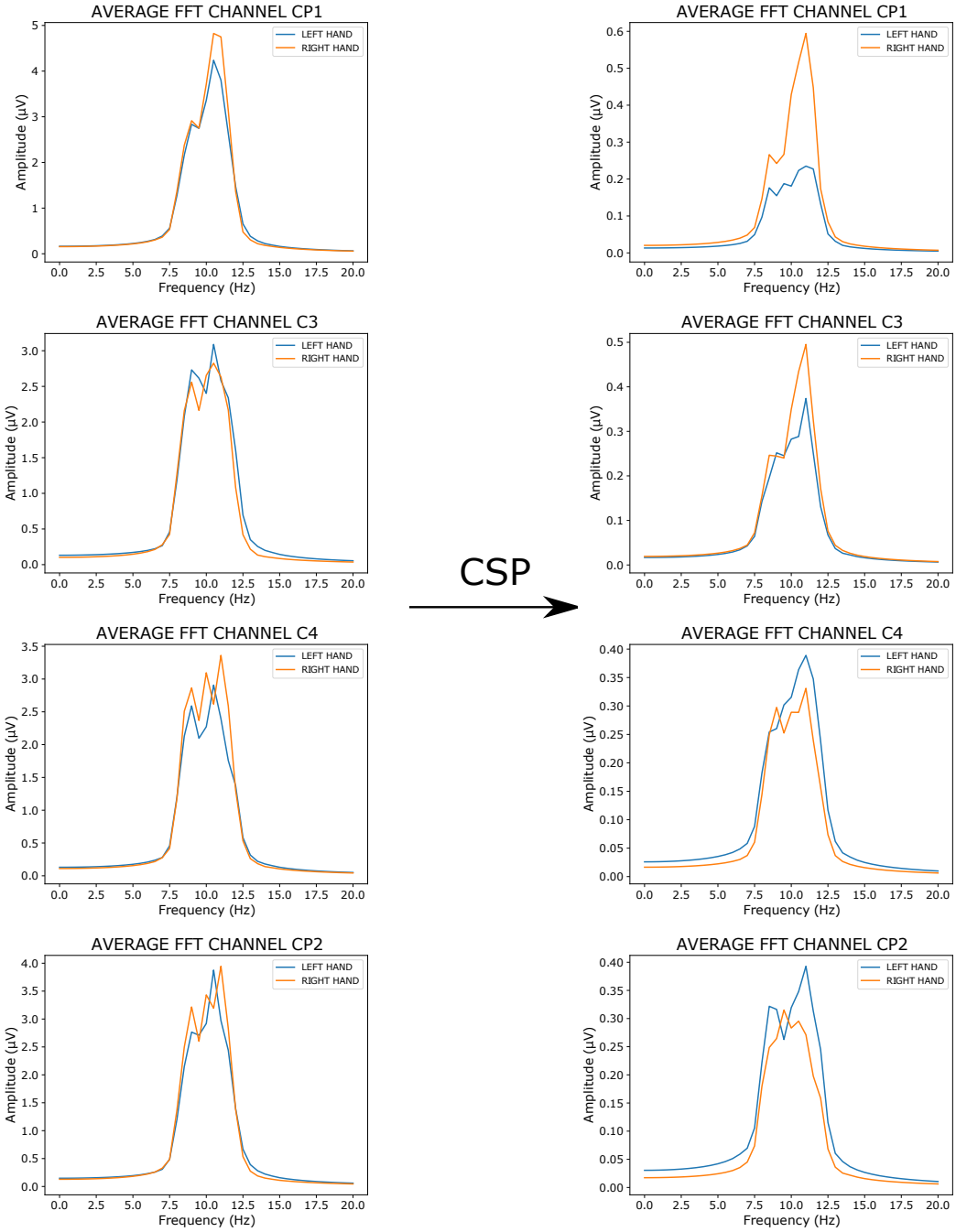


Figure 5.3: Visualization of ERD/ERS through the FFT before and after applying CSP in channel CP1, C3, C4 and CP2 for subject 2.

In fig. 5.3, the raw and CSP transformed FFT signals of four channels of subject 2 are shown. In the left column, there are arguably no apparent differences between the channels, but when looking at CP1, the left hand has lower amplitudes for around 10-12 Hz than right hand. However, after the signal is filtered with CSP, the differences are maximized and it becomes more apparent. This can be seen in the right column. The largest differences can be found in CP1, where left and right hand contrasts are highest. As right hand has highest amplitude for CP1 and C3, it seems to have an opposite effect than expected, indicating reversed ERD/ERS. It is not known why this happens. However, the CSP filter is trying to maximize differences in the tasks, which is clearly done.

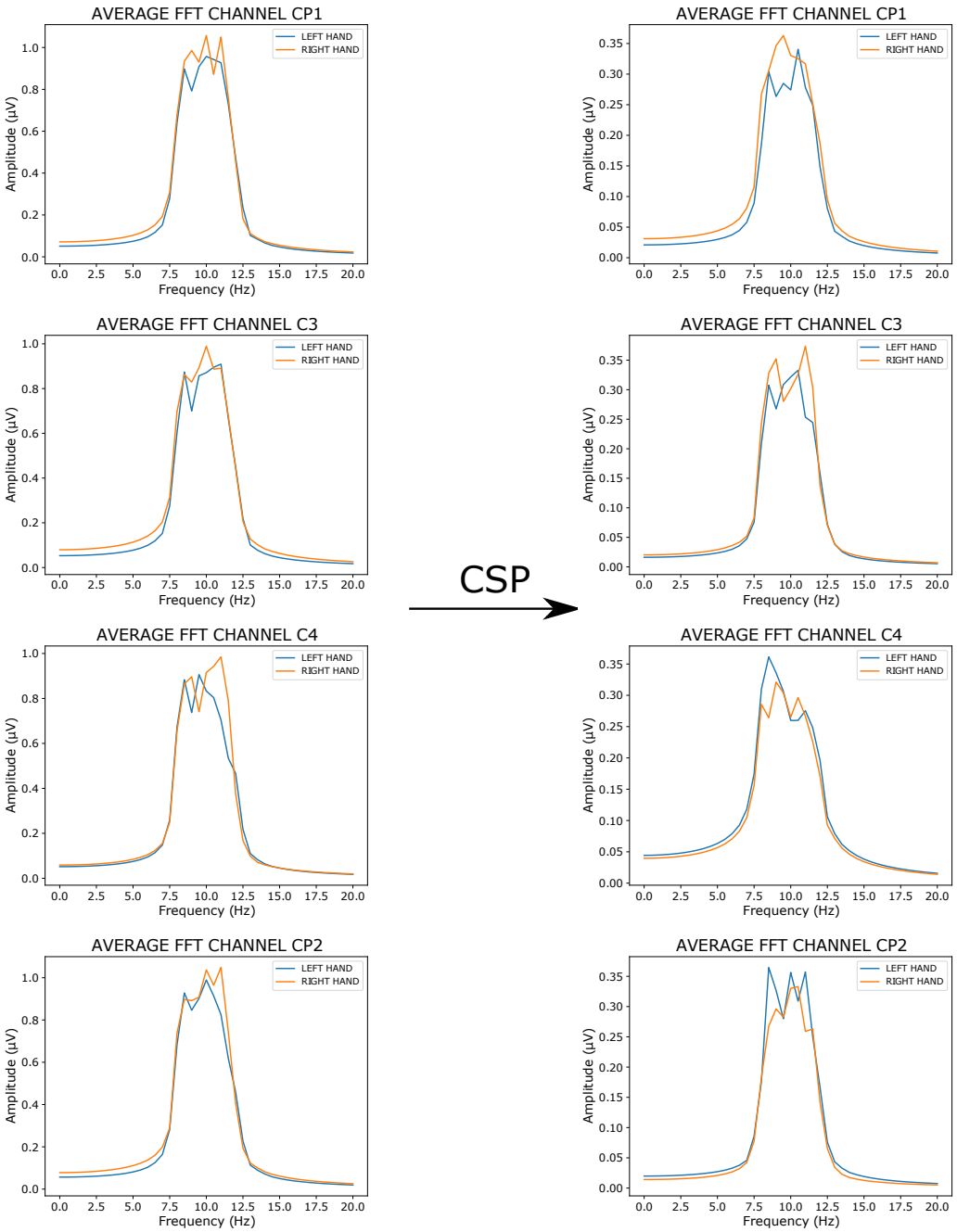
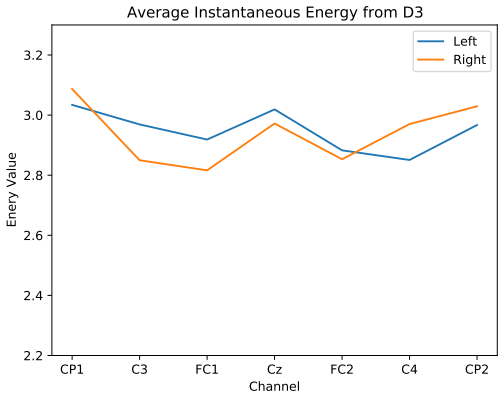


Figure 5.4: Visualization of ERD/ERS through the FFT before and after applying CSP in channel CP1, C3, C4 and CP2 for subject 7.

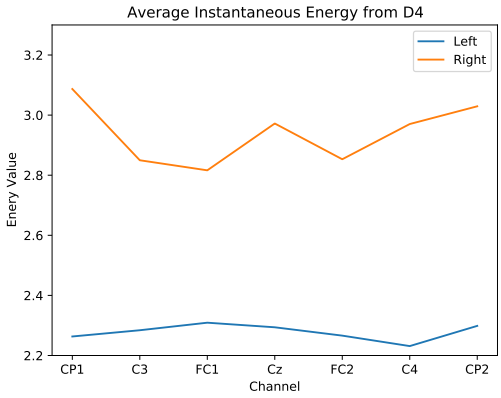
For subject 7, the difference in amplitudes is non-detectable. As seen in fig. 5.4, both right and left hand MI appears as having similar average amplitude. In contrast to subject 2, the CSP filtered signal does not ease the process of differentiating between the two classes. It appears that the method is not able to find any reliable patterns in the signal, and thus not being able to divide the classes. This may also explain the performance in the initial tests, whereas subject 7 got around 50% accuracy, while subject 2 got around 85% accuracy.

5.2.3 DWT Energy Features

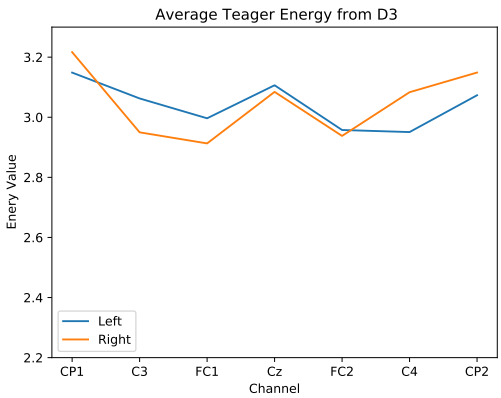
Similar to the visual analysis of CSP, a visual analysis of the extracted energy features for the DWT-based classifier is conducted for the same two subjects. In fig. 5.5, the average energy features from one session of subject 2 is extracted and plotted for each channel for both sub-bands used, D3 and D4. In the plots, the average energy level over one recording sessions is shown, to have a more general approach to the data.



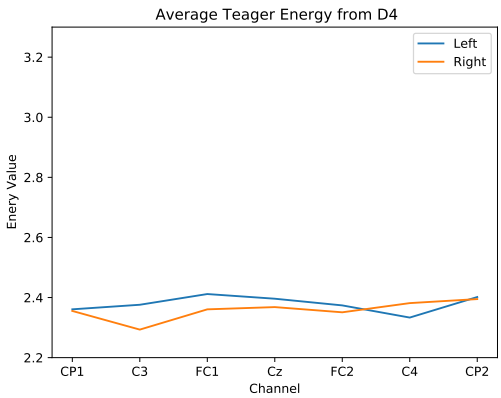
(a) Instantaneous Energy extracted from D3



(b) Instantaneous Energy extracted from D4



(c) Teager Energy extracted from D3



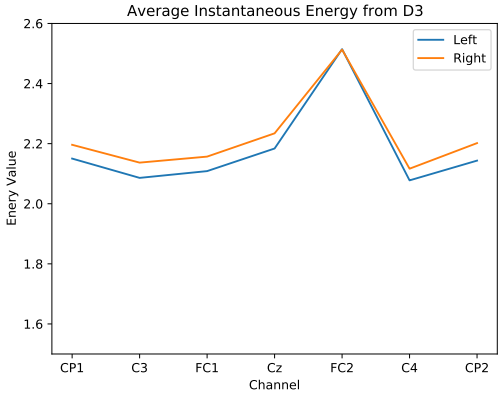
(d) Teager Energy extracted from D4

Figure 5.5: Visualization of ERD/ERS through energy features extracted from DWT levels D3 and D4 for subject 2

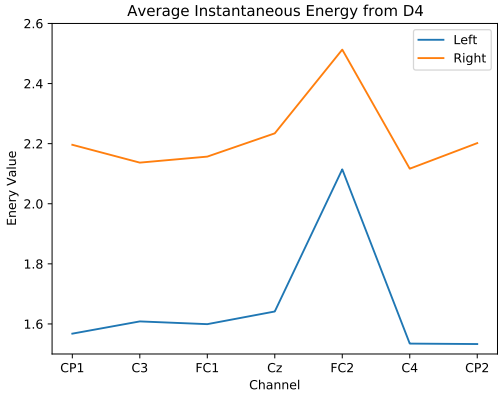
In fig. 5.5a, there are some small differences between the different MI classes. The two channels mostly exposed for alterations during MI according to theory, namely C3 and C4, seems to be oppo-

sites, as energies extracted during left hand movement seems to have relatively high values in C3 and low values in C4. In contrary, the right hand movement seems to result in a relatively low energy value in C3 and high value in C4. The same pattern can also be recognized in fig. 5.5c and fig. 5.5d, where the average teager energy for D3 and D4 are plotted. The instantaneous energy in D4, presented in fig. 5.5b, shows a general lower energy for left hand MI. This could correspond to focal ERD/surround ERS in the mu and beta bands.

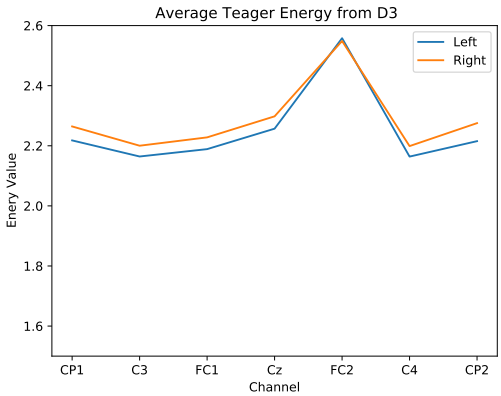
For subject 7, the average energy values extracted from D3 and D4 are presented in fig. 5.6. Here, the patterns in data seems less obvious than for subject 2. For the teager energy extracted from D3 and D4, as well as the instantaneous energy from D3 are all similar for both right and left hand MI. In these plots, the left hand MI is slightly lower than the energies from right hand MI. In fig. 5.5b, a similar pattern as for subject 2 can be spotted. Here, the left hand MI energies are quite a bit lower than the right hand MI ones.



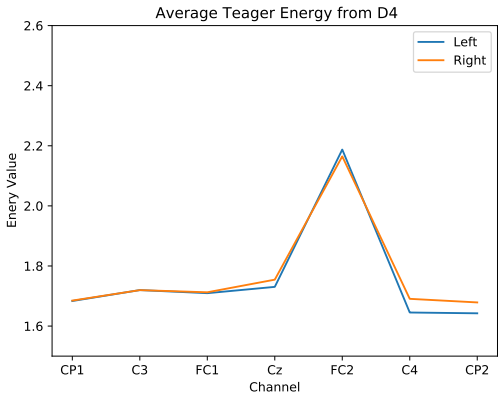
(a) Instantaneous Energy extracted from D3



(b) Instantaneous Energy extracted from D4



(c) Teager Energy extracted from D3



(d) Teager Energy extracted from D4

Figure 5.6: Visualization of ERD/ERS through energy features extracted from D3 and D4 for subject 15

From this visual analysis, it also seems to confirm the differences in classification accuracies. Bad performing subjects seems to not evoke any responses in the brain that can be picked up by the algorithms.

5.3 Evaluating Data of Three MI Classes

As mentioned earlier, the subjects six best-performing subjects were invited for a recording session with three classes. To create the best BCI drone controller, a high amount of classes while still having good accuracy is advantageous. Subject 1, 2, 6, 10, 15 and 16 was therefore invited for a recording session. However, there were technical issues during the recording with subject 6, and data with three MI tasks for subject 6 is therefore not considered further. For the remaining five subjects, an analysis similar to the one presented in section 5.1 was conducted, to gain insight in the different subjects' performance. However, this analysis is less comprehensive, only using CSP and SVM to get an idea of their performance. The results are presented in table 5.17 below. Here, the three MI tasks are classified, to see the ability to induce distinguishable responses in the brain when performing MI. This means an accuracy of 33.33% indicates randomness. Resting-state is not included.

Table 5.17: Accuracy in classification of left hand, right hand and foot MI for subjects 1, 2, 10, 15, 16, using CSP with SVM.

Method/Subject	**1**	**2**	**10**	**15**	**16**
CSP + SVM	54.17%	75.00%	39.17%	73.33 %	30.83%

From this simple analysis, it is clear that subject 2 and 15 obtained far greater accuracies than the others. Both subject 10 and 16 obtained around random accuracies, as a random classification accuracy now would be 33.33 %. Subject 1 did a decent run, but did arguably not obtain good enough results for a three class MI BCI.

5.4 Optimizing Methods and Parameters with NSGA-II

From the authors' work with their pre-projects and from literature studies in the early phases of the work with the thesis, several different methods for pre-processing, feature extraction and classification proved to be interesting. Therefore, the question of what data processing pipeline, i.e. which combination of methods to use, quickly arised.

There are a number of ways to approach this question. One way is to base the choice entirely on the literature, by choosing one interesting pipeline for pre-processing, feature extraction and classification that have proven to be successful in other studies. However, when many interesting methods are proposed in the literature, the idea of testing them all on your data is tempting. Since testing many combinations of methods manually and extensively is a life's work, the algorithm NSGA-II was used to search for combinations of preprocessing, feature extraction and classification methods giving better performance. Thus, the purpose of this experiment was to give a better basis to choose a good pipeline for the acquired data.

The data processing steps in which different methods or parameters were tested are as following:

1. *Re-referencing.* The re-referencing methods CAR and Laplacian referencing were tested, as well as no re-referencing.

2. *Bandpass filtering.* The purpose of this step was to extract the mu and beta bands. The following frequency bands were extracted and tested:

    - Mu (8-12 Hz). To isolate ERD in the mu band.
    - Mu and beta bands, separately (8-12 Hz and 12-30 Hz). To capture ERD in mu band and possible ERD/ERS in beta band.
    - Mu and beta bands, concatenated (8-30 Hz). To isolate ERD in mu and beta bands.
    - Mu and the central beta band (8-12 Hz and 15-19 Hz). To isolate ERD in mu band and capture the possible beta ERS in the central beta band.

3. *DWT.* As mentioned in section 3.1, the wavelet biorthogonal 2.8 with 4 levels of decomposition is used. In the combinations where the DWT was used, either all levels, i.e. all detail and approximation coefficients were used, or only levels D3 and D4. The levels D3 and D4 were chosen because they extract 15.63 to 31.25 Hz and 7.81 Hz to 15.63 Hz respectively, which approximately corresponds to the mu and beta bands.

4. *CSP.* In the feature extraction and classification pipelines where CSP was used, different numbers of components were tested, specifically 2, 3, 4, 5, 6, or the maximum of 7 components.

5. *Feature extraction and classification pipeline.* Nine different pipelines for feature extraction and classification were tested:

    - CSP with SVM (CSP, SVM)
    - CSP with LDA (CSP, LDA)
    - Covariance estimates with R-CSP and LDA (Cov, R-CSP, LDA)
    - FGDA and R-CSP applied to Covariance estimates and classification by LDA (Cov, FGDA, R-CSP, LDA)
    - Covariance estimates projected to the tangent space and classified with logistic regression (Cov, TS, LR)
    - Covariance estimates projected to the tangent space and classified with SVM (Cov, TS, SVM)
    - Covariance estimates filtered with FGDA and classification using MDRM (Cov, FGDA, MDRM)
    - With DWT, energy features were extracted and SVM was used for classification.

To represent different combinations of the above parameters and methods, 5 integer genes were used in each chromosome. NSGA-II was run with 25 chromosomes per generation. The optimization problems were set to terminate when the objective space change was less than 0.1 % accuracy in the last 15 generations, or when a maximum of 50 generations was reached. The objective space change criterion is calculated every 5^{th} generation.

The CNN was excluded from this experiment, due to neural networks being especially computationally expensive to train and having many hyperparameters. Additionally, the evaluation was carried out on relatively small datasets, while neural networks generally performs better with larger datasets [80].

Hierarchical Versus Flat Model

In addition to these combinations, two different classifier structures were tested and compared: Hierarchical and flat. We wanted to explore if using two different data processing pipelines in a two-level hierarchical structure increased the performance compared to a flat model.

The structure of the hierarchical classifier is shown in fig. 5.7. In this structure, the level 1 classifier was trained to classify the sample as either resting state or an MI task. If task was predicted, the level 2 model predicts one of the possible tasks, in this case among two or three tasks: Left hand versus right hand, or left hand versus right hand versus foot. This structure is proposed to investigate the possibility that some methods could be better fit for distinguishing the resting-state from active states, while other methods could be better at distinguishing the active states from another. If that is the case, combining two such methods in a hierarchical structure could increase accuracy.

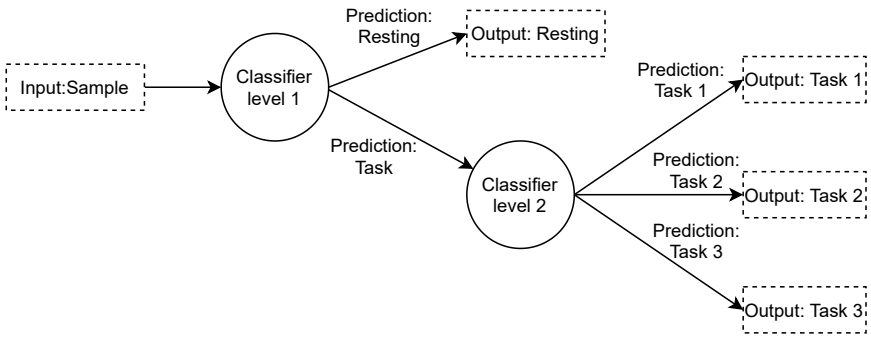


Figure 5.7: From an input to classification output with a hierarchical classifier.

To obtain the hierarchical model, NSGA-II was run twice, once with task data and once with resting and task data. Therefore, we will not only get one pipeline for feature extraction and classification for each level in the hierarchy, but also different parameters for the pre-processing steps, i.e. different combinations of re-referencing method, bandpass filters or DWT. In online classification, processing the data in two different ways is however not optimal in terms of run-time and may not be feasible, so if a hierarchical model is to be used, it may be necessary to find a common pre-processing pipeline for both levels. However, it can be interesting to see whether different pre-processing techniques better emphasize the characteristics of resting mental state and MI task execution.

In the flat structure, resting state and each of the MI tasks were used as classes. To obtain a model for the flat structure, NSGA-II was run once with each of the tasks and resting-state data.

5.4.1 Optimizing General Data Processing Pipeline for the 6 Best Subjects with data of Two MI Classes

In this experiment, the optimization problem was run with six objectives, specifically the cross-validation accuracy for each of the six subjects 1, 2, 6, 10, 15 and 16, when combining the data from the subject's four sessions with two MI tasks. The result of this optimization problem gave the opportunity to analyze which pipeline is the general best, i.e. giving as high as possible accuracy for all the six subjects simultaneously, or to find a best pipeline that considers all subjects but emphasizes the accuracy of certain subjects. Also, this analysis gave the opportunity to analyze the difference between the general best pipeline and the individual best pipeline. All of this is done by analyzing the pipelines in the pareto front.

Flat Model

To obtain a general best data processing pipeline for the six subjects, we first weigh each objective, i.e. each subject, equally by ranking the chromosomes in the pareto front, i.e. the best parameter and method combinations, by the the average of the obtained objective values. Then, the pareto front is plotted in a Parallel Coordinate Plot (PCP) where the obtained objective values of the four best chromosomes according to this ranking are highlighted. We first look at the results of the algorithm for the flat model, i.e. where resting state, left hand and right hand are treated as separate classes in a classification model. The four best chromosomes are highlighted in fig. 5.8. The corresponding parameter values, methods and mean accuracies are shown in table 5.18.

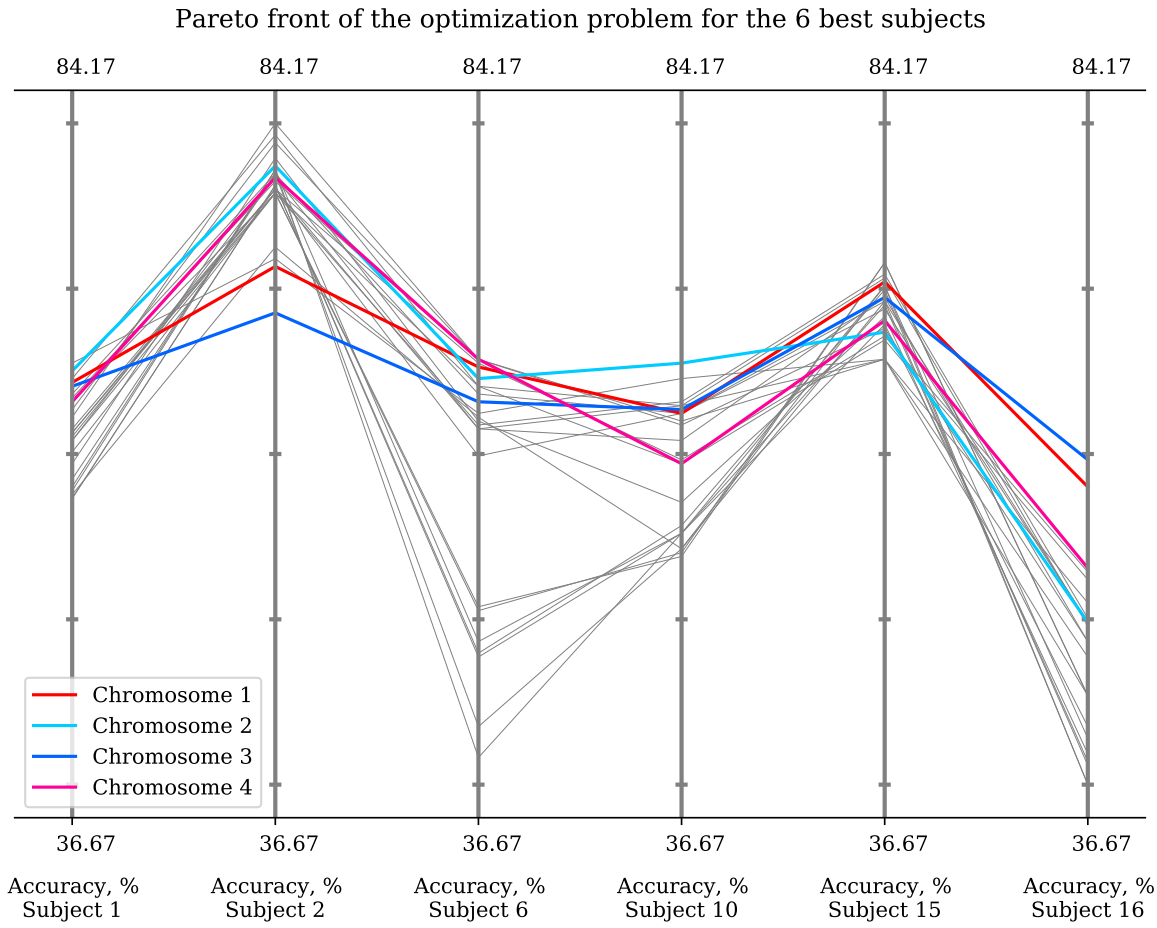


Figure 5.8: Resulting pareto front of the optimization problem for a flat model with subjects 1, 2, 6, 10, 15 and 16. The chromosomes obtaining the highest mean accuracies are highlighted. Three classes are classified: Left hand MI, right hand MI and resting-state.

Table 5.18: The re-referencing method, bandpass range, DWT levels, number of CSP components, and feature extraction and classification methods corresponding to each chromosome among the four highest mean accuracies for subject 1, 2, 6, 10, 15 and 16 in a flat model.

Chromo	**Re-ref**	**Bandpass range(s)**	**DWT lvls**	**CSP n.c.**	**FE and classification**	**Mean acc**
1	CAR	8-12 and 12-30 Hz	N/A	3	Cov, FGDA, R-CSP, LDA	66.71 %
2	Laplacian	8-12 and 12-30 Hz	N/A	N/A	Cov, TS, LR	66.30 %
3	CAR	8-12 and 12-30 Hz	N/A	N/A	Cov, FGDA, MDRM	65.88 %
4	Laplacian	8-12 and 12-30 Hz	N/A	6	Cov, FGDA, R-CSP, LDA	65.60 %

When only looking at table 5.18 and not considering fig. 5.8, since chromosomes 1 and 4 are only separated by 1.66 % mean accuracy, it is easy to think that there is not that much of a difference between the different re-referencing methods and the different number of components in CSP. However, when looking at fig. 5.8, it is clear that within some of the subjects there is a large difference in the accuracy with chromosomes 1 and 4.

Here, chromosome 1 is giving the highest mean performance for the subjects. However, chromo-

somes 3 and 1 are performing relatively bad for subject 2. Subject 2 and 15 are the best performing subjects, according to both section 5.1 and the accuracies in the PCP plot, so the pipeline should preferably give a good performance for these two subject as their data should be the easiest for an algorithm to classify. While chromosomes 2 and 4 perform better for subject 2, these two chromosomes perform relatively bad for subject 15. In this case, the need for choosing trade-offs in NSGA-II is clear - it is difficult to obtain a combination of methods that maximizes the performance of even just these two best subjects simultaneously.

From this, it is noted that chromosome 1 with the best mean accuracy could serve as one suggestion for a general combination of methods for several subjects, as it is difficult to find a chromosome that satisfy the desired trade-offs. The parameters and methods of chromosome 1 are CAR, 8-12 and 12-30 Hz, 3 CSP components, and the pipeline Covariances, FGDA, R-CSP and LDA, and it obtained a mean accuracy of 66.71 %.

Hierarchical Model

Now, it is attempted to obtain the general best data processing pipelines for the six subjects for a hierarchical model. Again, the chromosomes in the pareto front are ranked by the average of the corresponding objective values. It is reminded that this analysis must be done in two steps for the hierarchical model, as the optimization problem was run separately for the two levels. Here, both the results of a hierarchical model with one common pre-processing technique and a hierarchical model with two different pre-processing techniques are analysed here.

First, the results for the first level of the hierarchy, i.e. classification of resting state versus MI task execution is considered. Again, the four best chromosomes are highlighted in fig. 5.9. The corresponding parameter values, methods and mean accuracies are shown in table 5.19.

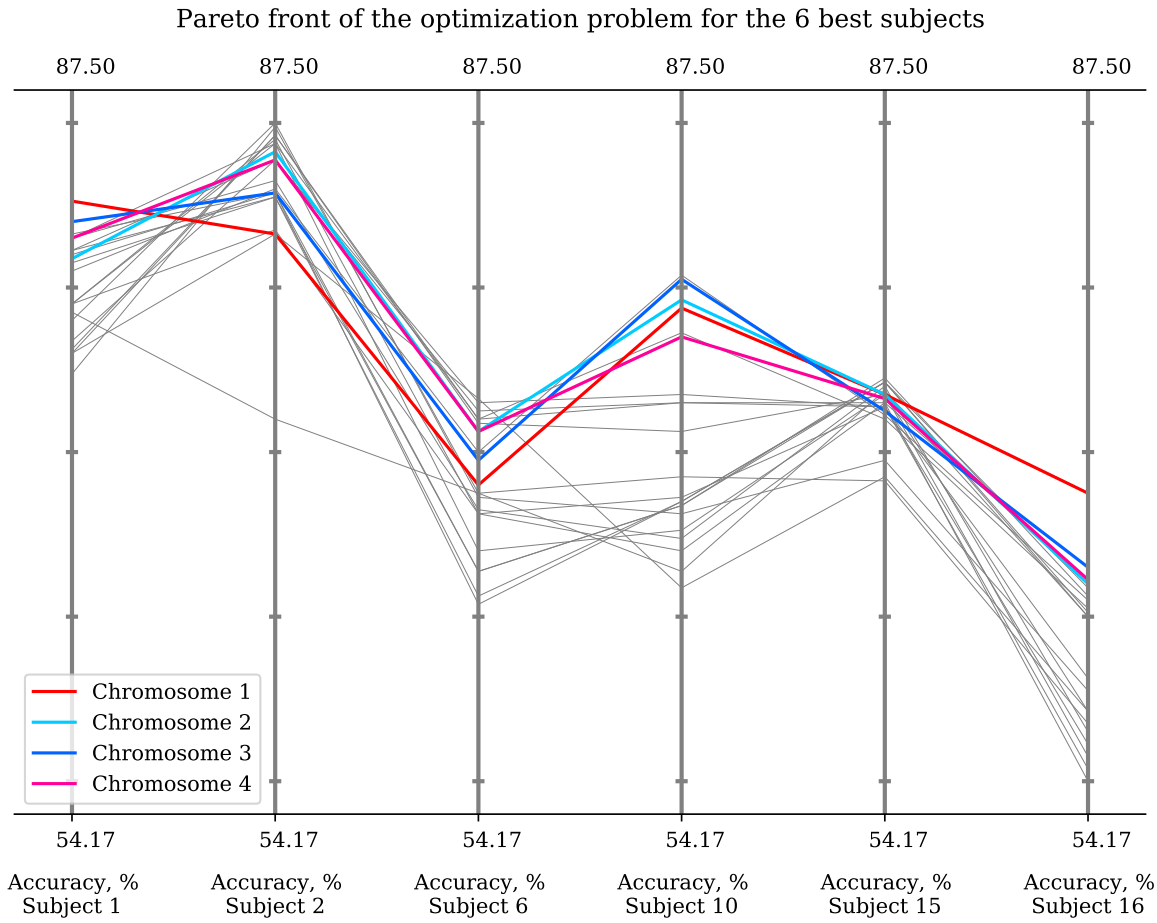


Figure 5.9: Resulting pareto front of the optimization problem for the first level of a hierarchical model with subjects 1, 2, 6, 10, 15 and 16. The chromosomes obtaining the highest mean accuracies are highlighted. Two classes are classified: MI or resting-state.

Table 5.19: The re-referencing method, bandpass range, DWT levels, number of CSP components, and feature extraction and classification methods corresponding to each chromosome among the four highest mean accuracies for subject 1, 2, 6, 10, 15 and 16 in the first level of a hierarchical model.

Chromo	Re-ref	Bandpass range(s)	DWT lvls	CSP n.c.	FE and classification	Mean acc
1	None	8-12 and 12-30 Hz	N/A	2	Cov, FGDA, R-CSP, LDA	75.87 %
2	Laplacian	8-12 and 12-30 Hz	N/A	N/A	Cov, TS, SVM	75.83 %
3	CAR	8-12 and 12-30 Hz	N/A	N/A	Cov, TS, LR	75.73 %
4	Laplacian	8-12 and 12-30 Hz	N/A	2	Cov, FGDA, R-CSP, LDA	75.63 %

Chromosome 1 (no re-referencing, 8-12 and 12-30 Hz bandpass, 2 CSP components, Cov, FGDA, R-CSP, LDA) has the highest mean accuracy in this case. Again, we note that the four best chromosomes are very close in mean accuracy, only differing by 0.24 % from the maximum to the minimum. However, the 2nd chromosome (Laplacian re-referencing, 8-12 and 12-30 Hz bandpass, Cov, TS and SVM) is giving equal or better performance than chromosome 1 for all subjects except subjects 1 and 16, who are the lowest ranked subjects regarding performance. Therefore, by choosing chromosome 2, the performances of the better subjects are emphasized more, while the mean accuracy only decreases by 0.04 %. This could be the better option if a general-best classifier in the two steps of the

hierarchical model were to be used in an application.

Now, the second level of the hierarchy, i.e. classification of left hand MI versus right hand MI is considered. The four best chromosomes are highlighted in fig. 5.10. The corresponding parameter values, methods and mean accuracies are shown in table 5.20.

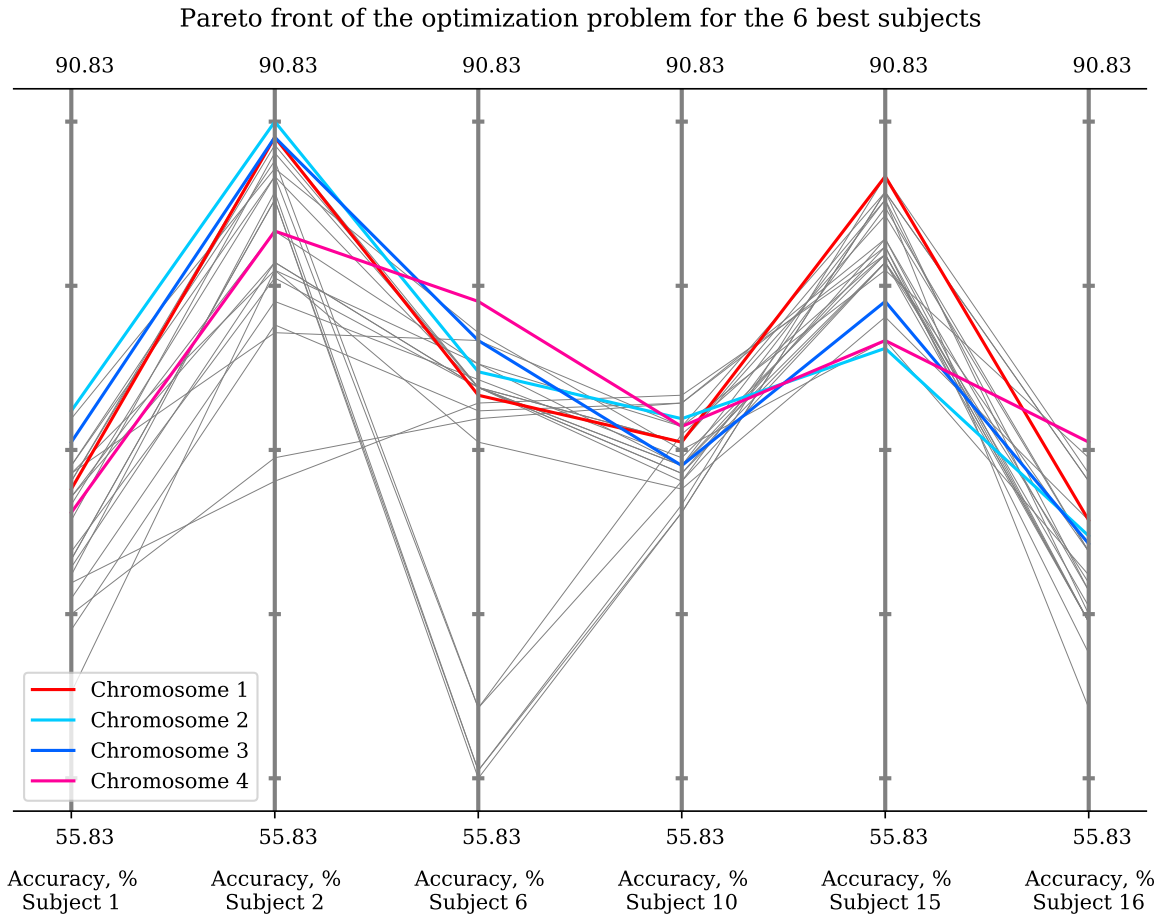


Figure 5.10: Resulting pareto front of the optimization problem for the second level of a hierarchical model with subjects 1, 2, 6, 10, 15 and 16. The chromosomes obtaining the highest mean accuracies are highlighted. Two classes are classified: Left hand MI or right hand MI.

Table 5.20: The re-referencing method, bandpass range, DWT levels, number of CSP components, and feature extraction and classification methods corresponding to each chromosome among the four highest mean accuracies for subject 1, 2, 6, 10, 15 and 16 in the second level of a hierarchical model.

Chromo	Re-ref	Bandpass range(s)	DWT lvls	CSP n.c.	FE and classification	Mean acc
1	CAR	8-30 Hz	N/A	N/A	Cov, TS, SVM	78.13 %
2	Laplacian	8-30 Hz	N/A	N/A	Cov, TS, LR	77.71 %
3	Laplacian	8-30 Hz	N/A	N/A	Cov, TS, SVM	77.50 %
4	Laplacian	8-12 and 12-30 Hz	N/A	4	Cov, FGDA, R-CSP, LDA	77.29 %

Here, we see that chromosome 1 generally gave a decent performance for all the subjects. The 2nd, 3rd and 4th best chromosomes all gave a relatively bad performance for subject 15 even though they gave good performance for the rest of the subjects. Again, a good performance for the best subjects 2 and 15 is preferred, which further supports the choice of the 1st chromosome as the best parameter combination.

Next, the data processing pipelines for both levels of the hierarchy are considered together. To compare the general hierarchical model to the general flat model, it is necessary to compute the accuracies from a hierarchical model composed of two of the best-performing chromosomes for each of the steps. To give the hierarchical classifier a fair chance against the flat model, the performance is maximized across subjects by choosing the best chromosome in each level in the hierarchy. Thus, the hierarchical classifier tested first is composed of the following methods and defined as:

- *Level 1 (rest versus MI task execution):* No re-referencing, 8-12 and 12-30 Hz bandpass filtering, 2 CSP components, Cov, FGDA, R-CSP, LDA.
- *Level 2 (left hand versus right hand MI):* CAR, 8-30 Hz bandpass, Cov, TS, SVM

The resulting mean performance for the above hierarchical model is 64.98 %, 1.73 % less than the flat model.

However, as already mentioned, using two different pre-processing pipelines for the two levels of the hierarchy is very ineffective in terms of run-time and may not be very beneficial in terms of accuracy. Therefore, one common pre-processing pipeline is also chosen, and corresponding accuracies computed, to see if such a model can compete with the flat model.

To obtain one common pre-processing pipeline, one can first consider the bandpass mode. Then, there are two options - either allowing the use of concatenated mu and beta bands in the first level of the hierarchy, or allowing the use of mu and beta bands separately for classification in the second level of the hierarchy. The latter is the alternative giving the smallest total loss of accuracy from the best chromosomes. In this case, one can use the 4th chromosome in the second level of the hierarchy. This chromosome uses Laplacian re-referencing, which means it shares both pre-processing steps with the 2nd best chromosome of the first level of the hierarchy - see table 5.20 and table 5.19. This gave a common pre-processing pipeline at the cost of a total of 0.88 % mean accuracy compared to using the 1st chromosome from both hierarchy levels. Thus, the second option for a hierarchical model is as following:

- *Pre-processing:* Laplacian re-referencing, 8-12 and 12-30 Hz bandpass filters
- *Level 1 (rest versus MI task execution) feature extraction and classifier:* Cov, TS, SVM
- *Level 2 (left hand versus right hand MI) feature extraction and classifier:* Cov, TS, SVM

The resulting performance for the above hierarchical model is 64.17 %, 2.54 % less than the flat model and only 0.81 % less than the hierarchical model with different pre-processing methods.

Result

In table 5.21, the results of the flat and hierarchical models are summed up.

Table 5.21: The mean accuracy across subjects 1, 2, 6, 10, 15 and 16 obtained by the flat and hierarchical models, when classifying left hand MI, right hand MI and resting-state.

	Mean accuracy
Flat model	66.71 %
Hierarchical model - differing pre-processing	64.98 %
Hierarchical model - common pre-processing	64.17 %

The method for average best performance of the six subjects with 2 MI tasks was a flat model with re-referencing by CAR, 8-12 and 12-30 Hz bandpass filters, feature extraction and classification by Cov, FGDA, R-CSP with 3 components and LDA, with an average accuracy of 66.71 %.

If we were to choose a combination of methods and parameters for real-time operation with an unknown subject, it could be beneficial to choose the combination of parameters and methods which gave the best mean. However, operating the drone with an unknown subject is not an aim in this work.

Although the differences in mean accuracy of the best chromosomes are very small, the differences in accuracy with the same chromosomes within a subject are generally large. This indicates that there is a potential of obtaining better performance for each subject by finding subject-specific data processing pipelines. This analysis could be carried out for all six subjects. However, it was also again confirmed that subjects 2 and 15 outperform the other subjects. Therefore, these two subjects were chosen as candidates for controlling the drone. To maximize the performance for the online operation of the drone, the results from this optimization problem are analyzed with respect to each of the subjects 2 and 15 in the further experiments. Then, the goal was to obtain a better, individual method for each of the subjects that later could be tested in real-time.

5.4.2 Optimizing Data Processing Pipeline for Subject 2 with Data of Two MI Classes

In this section, the results of the optimization problem in section 5.4.1 are analyzed with respect to subject 2 to obtain a higher accuracy for subject 2 than the subject-general pipeline could obtain. Again, the results of a flat model is first analyzed, then the results of a hierarchical model.

Flat model

With two MI tasks, the flat model classifies three classes: rest, left hand imagery and right hand imagery.

From the optimization problem in the previous experiment, the highest accuracy for subject 2 was obtained with Laplacian re-referencing, 8-30 Hz bandpass filter, and classification and feature extraction by Cov, TS, LR, with a score of 84.17 %. This is 9.83 % higher than the accuracy of subject 2 with the subject-general best combination, which is a significant improvement.

Hierarchical Model

With two MI tasks, the hierarchical model first classifies a sample as either rest or MI task, before it classifies the MI tasks as either left hand or right hand imagery. For the hierarchical model, the result of a hierarchical model with differing pre-processing pipelines is first found.

By choosing the best methods in classification of rest versus MI task and the best methods for classifying the MI task with respect to subject 2, the resulting hierarchy is:

- *Level 1 (rest versus MI task execution):* Laplacian re-referencing, 8-12 and 12-30 Hz bandpass filters, Cov, CSP with 6 components and LDA
- *Level 2 (left hand versus right hand MI):* Laplacian re-referencing, 8-30 Hz bandpass filter, Cov, TS, LR.

This gave an accuracy of 82.50 %, which is 1.67 % less than the flat, subject-specific model. The subject-general best hierarchical classifier obtained 76.39 % for subject 2, which means this subject-specific hierarchical classifier gave an improvement of 6.11 %.

Result

Both the flat and the hierarchical subject-specific models gave an improvement over the subject-general models, with 9.83 % and 6.11 %, respectively. The flat model achieved an accuracy of 84.17%, while the hierarchical model achieved an accuracy of 82.50%. Thus, the flat model performed better than the hierarchical model with differing pre-processing techniques in each level. A hierarchical model with one common pre-processing technique will not perform better than this hierarchical model when only considering one metric. Therefore, because the the hierarchical model with differing pre-processing techniques performed worse than the flat model, it is not necessary to carry out the analysis of hierarchical model with common pre-processing pipelines.

The best method for subject 2 with 2 MI tasks and a resting-state was thus a flat model with Laplacian re-referencing, 8-30 Hz bandpass filter, feature extraction and classification by Cov, TS and LR, with an accuracy of 84.17 %.

5.4.3 Optimizing Data Processing Pipeline for Subject 15 with Data of Two MI Classes

Here, the results of the optimization problem in section 5.4.1 are analyzed with respect to subject 15, to obtain a subject-specific method for the subject. First, the result of a flat model is presented, before the results of a hierarchical model.

Flat Model

The highest accuracy for subject 15 was obtained with CAR or with no re-referencing, 8-30 Hz bandpass filter, and CSP with 7 components with SVM, with an accuracy of 74.17 %. This is an improvement of 1.39 % from the 72.78 % achieved with the general best combination. This improvement is smaller than the improvement of subject 2, which can be confirmed by chromosome 1 in fig. 5.8.

Hierarchical Model

For the hierarchical model, we first find the results of a hierarchical model with differing pre-processing pipelines.

The resulting hierarchy with differing pre-processing when choosing the best methods for each level is:

- *Level 1 (rest versus MI task execution):* Re-referencing using CAR, 8-12 and 12-30 Hz bandpass filters, CSP with 7 components, SVM.
- *Level 2 (left hand versus right hand MI):* Re-referencing using CAR, 8-30 Hz bandpass filtering, Cov, TS, SVM.

This gave the accuracy 69.17 %. This is 5.00 % less than the flat, subject-specific model. The loss of accuracy from the flat to the hierarchical model is bigger here than with subject 2. Furthermore, the subject-general hierarchical classifier obtained 68.61 % for subject 15, which means the subject-specific hierarchical classifier gave an improvement of only 0.46 %.

Result

Again, both the flat and hierarchical subject-specific models were better than subject-general models, with an increase of 1.39 % and 0.46 % respectively. Thus, the increase was notably smaller for subject 15 than for subject 2. This could be predicted from the results in section 5.4.1, as it was noted that the subject-general best methods had a relatively bad performance for subject 2.

The flat model performed 5.00 % better than the hierarchical model. Thus, it is not necessary to choose a hierarchical model with one common pre-processing method. The best method for subject 15 with 2 MI tasks was thus a flat model with no re-referencing, 8-30 Hzbandpass filter, feature extraction and classification by CSP with 7 components and SVM, with an accuracy of 74.17 %.

5.4.4 Optimizing Data Processing Pipeline for Subject 2 with Data of Three MI Classes

As can be seen from section 5.3, subject 2 and 15 also outperformed the other subjects with three MI tasks. Therefore, a subject-specific method for subject 2 with three MI classes is found in this section.

To do this, it was necessary to run a new optimization problem with NSGA-II, as the previous optimization problem was run for data with two MI tasks. Now, NSGA-II was run specifically for subject 2, with three MI tasks, both for hierarchical and flat models. As only two sessions were recorded with three MI tasks, the optimization objectives for the corresponding optimization problems are as follows: Cross-validation accuracy in session 1, session 2, and the cross-validation accuracy for the the accumulated data from sessions 1 and 2.

Then, the results in the pareto front are considering each of the sessions individually. However, the choice of method from the pareto front is based entirely off the last objective, the accuracy for the accumulated data. Partly because this makes the analysis easier, but most importantly because this

is the data that will be used for training the algorithms before a possible online classification. It is also more important that the final method is good at generalizing the patterns, rather than good at the individual sessions.

Flat Model

With three MI tasks, the flat model classifies a sample as either rest, left hand imagery, right hand imagery or foot imagery.

From the new optimization problem with three classes, the highest accuracy for the accumulated data from sessions 1 and 2 was 76.88 %, obtained with re-referencing by CAR, 8-12 and 12-30 Hz bandpass filters, Cov, FGDA, R-CSP with 6 components and LDA. It is interesting that unlike in earlier experiments where the range in accuracy of the four best alternatives typically is around 0.5-1%, the best alternative is in this case 4.38 % better than the 2^{nd} alternative, and the range of the four best methods is 5.63 %. Also, the best method included use of 8-12 and 12-30 Hz bandpass filters, while the three subsequent methods included the use of a 8-30 Hz bandpass filter.

Hierarchical Model

With three MI tasks, the hierarchical model first classifies a sample as either rest or MI task, before it classifies the MI tasks as either left hand, right hand or foot imagery. First, a hierarchical model with differing pre-processing methods is obtained. When choosing the best methods for each level, the resulting hierarchical model is composed of the following methods:

- *Level 1 (rest versus MI task execution):* No re-referencing, 8-12 and 12-30 Hz bandpass filters,CSP with 2 components and classification by LDA.
- *Level 2 (left hand versus right hand versus foot MI):* No re-referencing, 8-30 Hz bandpass filter, CSP with 4 components, classification by SVM.

This gave an accuracy of 63.61 %. This is as much as 13.27 % less than the flat model.

Result

As the flat model performed so much better than the hierarchical model, it was not necessary to analyse a hierarchical model with a common pre-processing method. The best method for subject 2 with 3-task data was thus a flat model with re-referencing by CAR, 8-12 and 12-30 Hz bandpass filters, Cov, FGDA, R-CSP with 6 components and LDA, which obtained an accuracy of 76.88 %.

5.4.5 Optimizing Data Processing Pipeline for Subject 15 with Data of Three MI Classes

Here, a subject-specific method for subject 15 is found, based on a subject-specific optimization problem with NSGA-II. This procedure was exactly the same as for the 3-task data of subject 2 in section 5.4.4, with the same optimization problems and the same objectives, the only difference being the subject that the data belongs to. Again, the pareto front of the optimization problem is analyzed with respect to the accuracy on the accumulated data from sessions 1 and 2.

Flat Model

The highest accuracy obtained for a flat model was 67.50 %, with Laplacian re-referencing, 8-30 Hz bandpass filter, and feature extraction and classification by Cov, TS, and SVM.

Hierarchical Model

In this case, when choosing the best methods for each level of the hierarchy, the same pre-processing methods are actually chosen in both levels, which means a common pre-processing method is obtained. Now, the hierarchical model consists of the following methods:

- *pre-processing:* No re-referencing, 8-12 and 12-30 Hz bandpass filters.
- *Level 1 (rest versus MI task execution):* Cov, FGDA, R-CSP and LDA. R-CSP could be used with an arbitrary number of components (2-7), this did not affect the result.
- *Level 2 (left hand versus right hand versus foot MI):* Cov, FGDA, MDRM.

This is the first time the separate filtering of mu and beta bands (8-12 and 12-30 Hz bandpass filters) have been the best-performing method for task classification. In previous cases, the 8-30 Hz bandpass filter has typically dominated the top three best-performing methods for the second level of the hierarchy. Moreover, in addition to the number of components in the first level not affecting the result, the same result was also achieved with or without re-referencing by CAR, in both levels. Then, it is better to choose no re-referencing, at is avoids the (rather quick) computation of re-referencing and therefore is *slightly* faster.

This model obtains an accuracy of 63.75 %, which is 3.75 % less than the flat model.

Result

Again, the flat model performed better than the hierarchical model, this time by 3.75 %. The best method for subject 15 with 3 MI tasks was thus a flat model with Laplacian re-referencing, 8-30 Hz bandpass filter, feature extraction and classification by Cov, TS, and SVM, with an accuracy of 67.50 %. This is 9.38 % less than the accuracy achieved by subject 2 with 3 MI tasks.

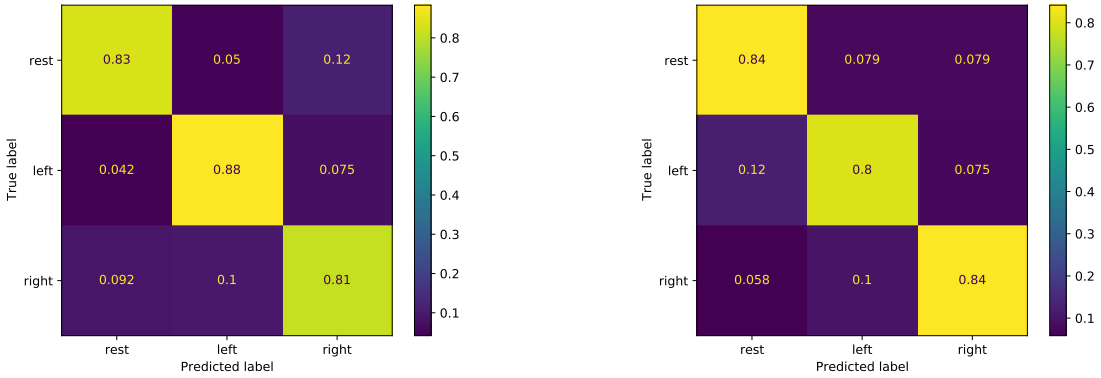
5.4.6 A More In-depth Comparison of the Flat and Hierarchical Models

In order to properly evaluate the differences in the flat and hierarchical structures, other than in regards of accuracy, confusion matrices are used. Here, the sensitivity of each class can be seen, as well as how the false negatives are distributed across classes. Sensitivity in certain classes may be more important in some systems. Also, it is necessary to ensure that the final, selected methods are recognising all classes equally well, meaning that the sensitivity should be somewhat even for each of the classes.

Flat and Hierarchical Model Comparison for Subject 2

In fig. 5.11, the flat and hierarchical confusion matrices are shown for subject 2. They both appear to perform similarly, with an accuracy of 84.0% and 82.7% respectively. However, there are a few

differences in MI sensitivity. In the flat model, left hand MI has higher sensitivity, 8% better than the hierarchical model. On the contrary, the hierarchical model is a bit better than the flat model for both rest and right hand MI.

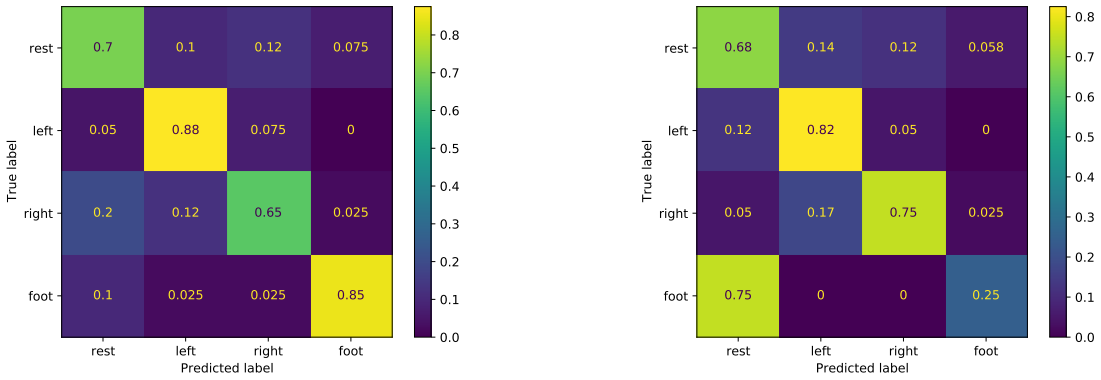


(a) Flat classification model

(b) Hierarchical classification model

Figure 5.11: Confusion matrices for the best flat and hierarchical model of subject 2 for two MI classes and a resting-state.

The confusion matrices for hierarchical and flat classification for three MI tasks and resting-state is shown in fig. 5.12. There is a clear distinction between the two matrices, as the hierarchical model predicts the foot task as rest 75% of the time. This is a large error that is not occurring in the flat model, making the flat model far more reliable than the hierarchical model.



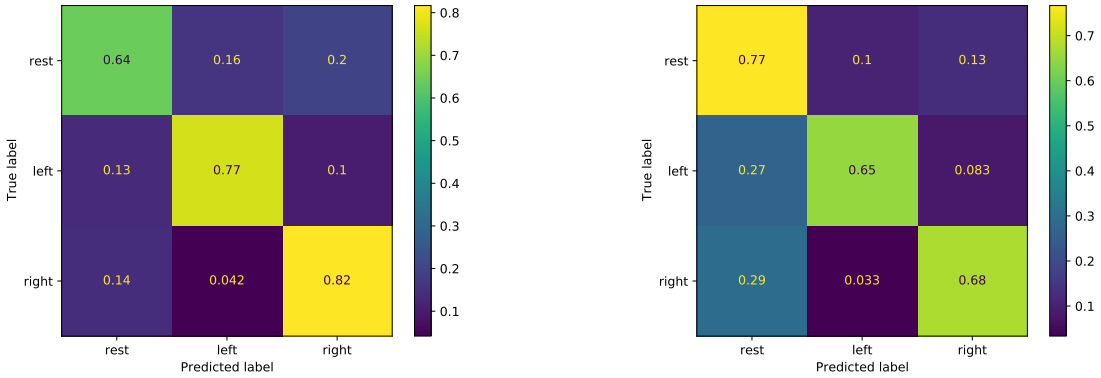
(a) Flat classification 3 classes

(b) Hierarchical Classification 3 classes

Figure 5.12: Confusion matrices for the best flat and hierarchical model of subject 2 for three MI classes and a resting-state.

Flat and Hierarchical Model Comparison for Subject 15

For subject 15, the differences between the hierarchical model and the flat model are more distinctive for two MI classes and resting-state. The hierarchical model has higher sensitivity for rest, whereas the flat model is better for the MI tasks. This can be beneficial, as it can be better to wrongly classify a task as rest, rather than classifying it as the wrong MI task.

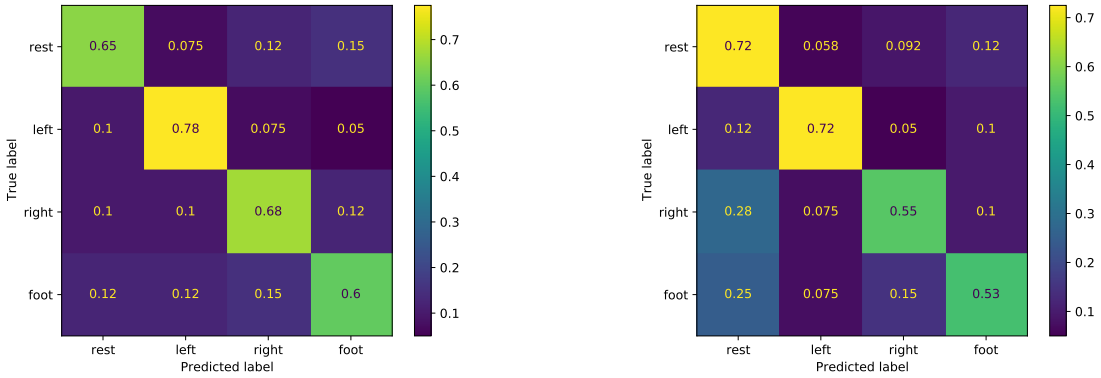


(a) Flat classification model

(b) Hierarchical classification model

Figure 5.13: Confusion matrices for the best flat and hierarchical model of subject 15 for two MI classes and a resting-state.

For three MI classes and resting-state, the problem appearing for subject 2 does not occur. However, there are some differences other than accuracy that can be noted for the two different models. The hierarchical model has worse accuracy, but when obtaining false negatives for MI tasks, it predicts rest more often. This is quite similar to the two MI task hierarchical model, and might be advantageous. However, the total higher accuracy of the flat model means this model is still chosen for further consideration.



(a) Flat classification for 3 classes

(b) Hierarchical classification for 3 classes

Figure 5.14: Confusion matrices for the best flat and hierarchical model of subject 15 for three MI classes and a resting-state.

5.4.7 Conclusion on Methods to Use for Subject 2 and 15

The best performances were achieved using subject-specific pipelines with a **flat structure**. The pipelines achieving the best performance for subjects 2 and 15 with 2 or 3 MI tasks are presented in table 5.22. **These pipelines are used for the two subjects for the rest of the experiments.**

Table 5.22: The best pipelines for subject 2 and 15, with 2 or 3 MI tasks. These are the pipelines that are used from now on.

Subject	MI tasks	Re-referencing	Bandpass filter(s)	Feature extraction and classification
2	2	Laplacian	8-30 Hz	Cov, TS, LR
2	3	CAR	8-12 and 12-30 Hz	Cov, FGDA, R-CSP with 6 components, LDA
15	2	None	8-30 Hz	CSP with 7 components, SVM
15	3	Laplacian	8-30 Hz	Cov, TS, SVM

5.4.8 Overall Analysis of the Results from NSGA-II

The flat structure outperformed the hierarchical structure in all cases, so it seems that the hierarchical structure did not give any benefit in these experiments.

For the bandpass filters, the 8-30 Hz and the 8-12, 12-30 Hz filters dominated the pareto front in all cases. The other bandpass filters, 8-12 Hz and 8-12, 15-19 Hz did never appear in the top four chromosomes, and rarely in the pareto front. Thus, it seems that the information in the beta band was important to include, but the central beta band (15-19 Hz) did not extract the most discriminating information.

There was no clear trend as to which re-referencing method was better. It depended on the subject and method for feature extraction and classification. In many cases, changing the re-referencing method only changed the performance with 0-2 %. In one case, however, going from no re-referencing to Laplacian in a pipeline gave a change of -3.75 % in one subject and +4.37% in another. In another pipeline, changing from CAR to Laplacian gave a change of +6.67% for one subject, and little to no change for the other subjects.

Additionally, a clear trend was not seen in the number of CSP components. Such a trend could possibly have been established with more data and a more thorough analysis.

In the total of 9 optimization problems ran, the DWT-based methods only appeared twice in the pareto front. This reflects the findings in the initial evaluation of the subjects in section 5.1, specifically the DWT-based method often performed significantly worse than the other methods. Both times DWT appeared in the pareto front was in classification of rest versus MI tasks, i.e. in the first level of the hierarchical structure. In both cases, only the levels D3 and D4 were extracted. This confirms the fact that the interesting information appear in the mu and beta bands, which D3 and D4 approximately correspond to, and that the other levels derived from the DWT are not interesting.

5.5 Cross-session Classification

For a BCI to work in real time, it is also crucial that the classifier is able to classify across sessions. There is a possibility for differences in data received from the Open-BCI headset, as it can be difficult to replicate the same exact position, scalp contact and other possible interferences. The classifier

needs to be able to handle such differences, as a calibration or creating new datasets for every live-run in undesirable.

A new test regime was created to test the capabilities of the classifiers to accurately predict classes on new sessions. For this regime, the classifier was first trained on three sessions and tested on the last for Dataset A. For Dataset B, the classifier trained on one session and tested on another, as only two sessions were available. These tests were carried through for the best classifiers of subject 2 and 15.

5.5.1 Classification Across Sessions for Subject 2

The results from the cross-session test for two MI tasks and resting-state is presented in table 5.23. The accuracies are all decent, ranging from 74.44% to 86.67%, where testing on session 4 scores highest. These are all good results indicating a real-time drone session can be successful.

Table 5.23: Cross session classification of two MI classes and resting-state for subject 2, using the individual best pipeline.

Training Session	Testing Session	Accuracy
2,3,4	1	75.56%
1,3,4	2	74.44%
1,2,4	3	83.33%
1,2,3	4	86.67%

For three MI classes and resting-state an accuracy of only 63.75% was obtained for both sessions. This is presented in table 5.24. There is a possibility that the low amount of data for three MI tasks can reduce the cross-session accuracy. The lower accuracies can also indicate that an online classification can become more difficult.

Table 5.24: Cross session classification of three MI classes and resting-state for subject 2, using the individual best pipeline.

Training Session	Testing Session	Accuracy
1	2	63.75%
2	1	63.75%

5.5.2 Classification Across Sessions for Subject 15

For subject 15, the cross session test for two MI tasks and resting-state performed well, as shown in table 5.25. The worst performance was for session 2, which granted an accuracy of 63.33%, while session 4 got an accuracy of 86.67%. The large span in these accuracies may indicate a decreased robustness. However, it may also be because of a bad performance during that exact session, which was mentioned in section 5.1.

Table 5.25: Cross session classification of two MI classes and resting-state for subject 15, using the individual best pipeline.

Training Session	Testing Session	Accuracy
2,3,4	1	70.00%
1,3,4	2	63.33%
1,2,4	3	73.33%
1,2,3	4	86.67%

The two sessions with three MI tasks and resting-state also had varying performances for the cross-session test. The results are presented in table 5.26. When testing on the second session, it scored 12.50% better than when testing on the first session. The lower accuracies may suggest that it is not suited for online classification.

Table 5.26: Cross session classification of three MI classes and resting-state for subject 15, using the individual best pipeline.

Training Session	Testing Session	Accuracy
1	2	68.75%
2	1	56.25%

5.6 Simulation of Online Classification

As a last experiment before connecting the entire system together and flying the drone, a simulation of the online classification is tested. For this experiment, one session is used as "real time data", while the rest of the data is used for training the classifier. The blocking state described in the state machines in section 4.6 is in use during this experiment.

Now, the entirety of the mock real-time data is used for classification. For this, every point of the data is now labeled with the belonging class. Then, the data is extracted 500 points at the time, as shown in fig. 4.8. This will introduce a changing offset between the time windows of the true labels and the predicted labels. Therefore, the predictions are evaluated as following: If the time window of a predicted task partly overlaps with the true task, and the correct task was predicted, this is regarded as a correct prediction. If resting-state or the wrong task is predicted in both time windows overlapping with the task, one error is counted. If a task is predicted during the time windows purely overlapping with the true rest, it is counted as an error. This is illustrated in fig. 5.15.

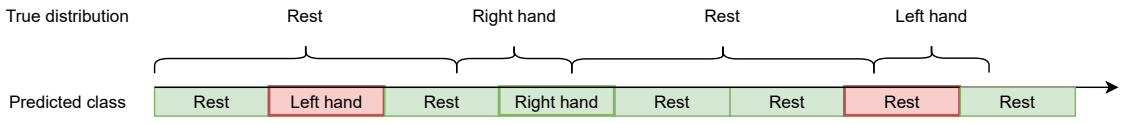


Figure 5.15: Illustration of evaluation of the predictions during simulation of online classification.

This experiment was performed for both subject 2 and 15, for two and three MI task data. For two MI classes, there are 30 trials of each MI class, and 168 trials of the rest class. For three MI classes, there are 20 trials of each MI class and 168 trials of the rest class. As there are so many more resting-state instances, the total accuracies presented are balanced, such that all classes are weighed the same. The accuracies of the different classes are presented in confusion matrices.

5.6.1 Simulation of Online Classification for Subject 2

For subject 2, the results from the two MI task data is presented in fig. 5.16. With a total accuracy of 84.85%, this is a good performance for an online simulation. There are rarely any guesses wrong between the two MI tasks, as it is mostly predicted rest when the prediction is wrong. Also, the rest class i predicted correctly 95% of the time. This is promising results regarding the online classification.

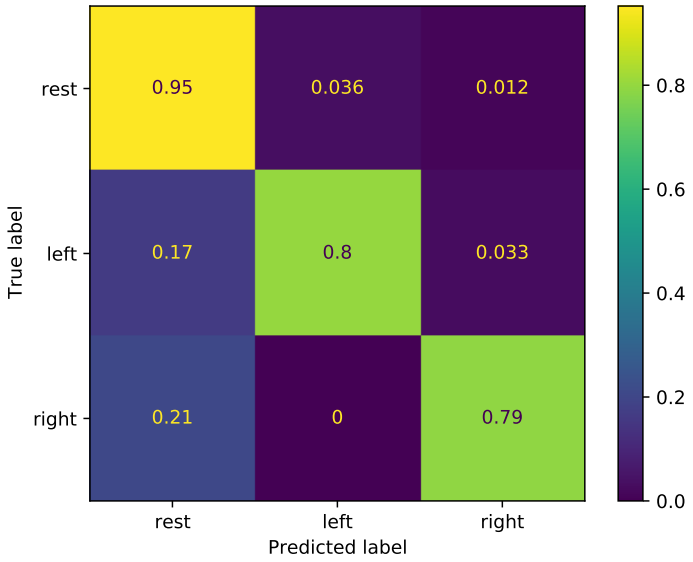


Figure 5.16: Confusion matrix based on the results of the simulated online session for subject 2 with two MI classes and a resting-state.

For the three MI task dataset, the results are worse. As seen in fig. 5.17, all MI classes have low performance, as the classifier often predicts these classes as the rest class instead. It only scored a balanced accuracy of 52.78%.

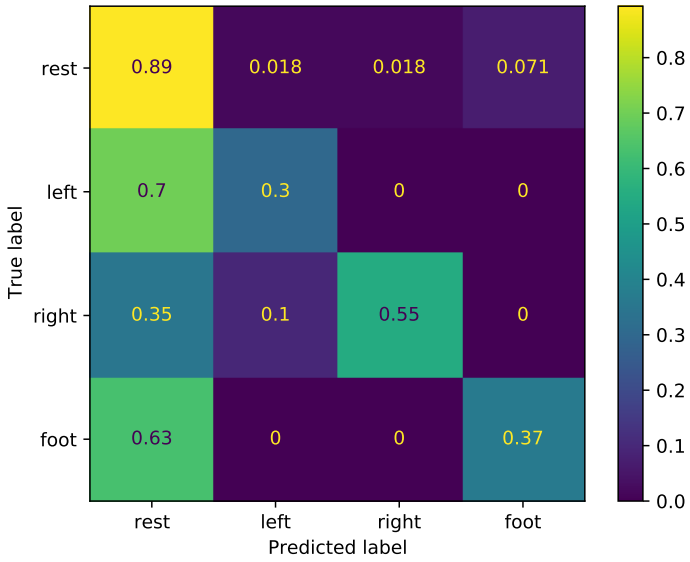


Figure 5.17: Confusion matrix based on the results of the simulated online session for subject 2 with three MI classes and a resting-state.

5.6.2 Simulation of Online Classification for Subject 15

For subject 15, the results from the session with two MI task and resting-state is presented in fig. 5.18. Similarly to subject 2, the rest sensitivity is high, at 93%. However, the MI task sensitivities are not as good, scoring 56% for left hand and 69% for right hand. This results in an overall accuracy of 73%, which is generally good.

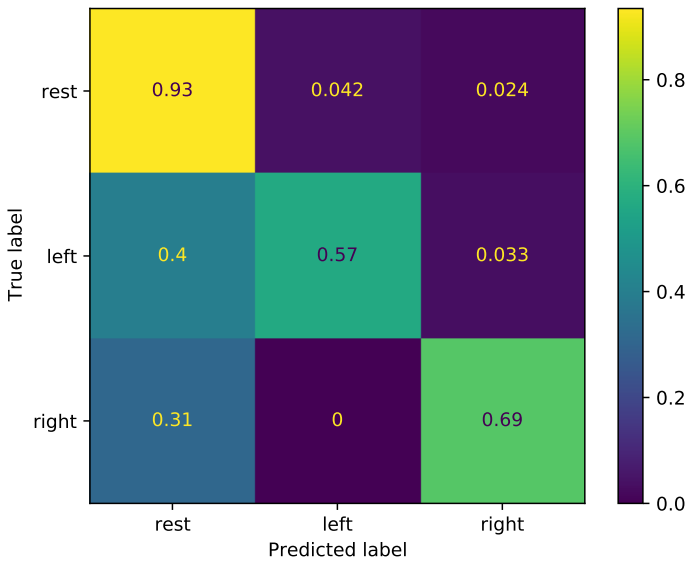


Figure 5.18: Confusion matrix based on the results of the simulated online session for subject 15 with two MI classes and a resting-state.

For the dataset with three MI tasks, the results from the experiment are shown in fig. 5.19. With a

total accuracy of 59.64%, subject 15 performed better than subject 2, even though the earlier results suggested that subject 2 would perform better. However, this is still not a great result. The foot class is rarely predicted correctly, as rest is chosen instead.

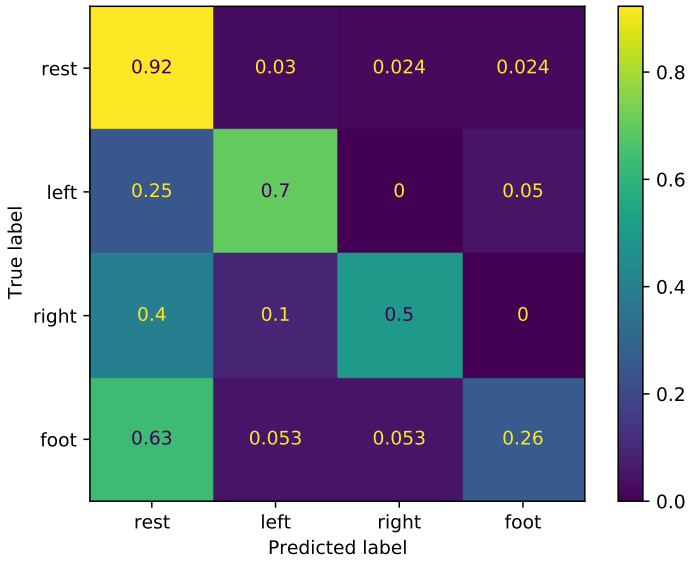


Figure 5.19: Confusion matrix based on the results of the simulated online session for subject 15 with three MI classes and a resting-state.

5.6.3 Summary

As both subject have acquired good results for 2 MI classes, both will be testing the drone in real time. This is presented in the next section. The three MI class system had worse score, and will therefore not be tested with the drone. However, it is tested online without the drone to further investigate the feasibility.

5.7 Online Classification with Drone Actuation

Subjects 2 and 15 were invited for the online experiments. This experiment was performed approximately 4 weeks after the training data was recorded. The models were trained on all training data from the previously recorded data - 4 sessions for two MI tasks, and 2 sessions for three MI tasks. For this experiment, the subjects were given sequences of commands to give to the drone. The inclusion of the blocking state in the state machine also makes it necessary to stay resting for at least one time segment between each command. The subjects were attempting to execute one of the following command sequences several times:

1. forward - forward - rotate - rotate - forward - rotate
2. forward - rotate - forward - rotate - forward - rotate
3. rotate - rotate - forward - forward - rotate - forward

The subjects were seated in front of the computer screen and could then view the feedback showing the drone command output. The subjects were told to proceed through the sequence as if no errors occurred.

For unknown reasons the drone was frequently drifting in an unpredictable manner, unrelated to the commands that were given. To exclude the possibility that the drifting could be related to the BCI system, the drone was also tested with a hard-coded sequence of commands using the same software interface. As the drifting also occurred in this case, it was assumed that it was unrelated to the BCI system, and rather related to damages to the drone or issues with the firmware, the latter being out of the scope of this work.

This problem made it difficult to complete a whole sequence, as the drone in most attempts drifted into the barrier netting at the edges of the experiment field before the sequence was finished, making it necessary to cancel the attempt. This could happen regardless of the starting position. Therefore, the results of the online experiments are separated in two for each subject: Attempts at the sequences where the drone is actuated, and attempts at the sequences while the drone is inactive, i.e. the commands are not sent to the drone, only output to a screen. The latter was included in the experiment to get results from longer sequences than the drifting drone would allow. The distinction in the results is made due to the possibility that the drone being active could affect the mental state of the subject, and thus affect the performance.

The performance is measured according to the metrics mentioned in section 4.7, i.e. TPR and FPR. Due to having limited time available with each subject for the online exiperiments, a low number of attempts were conducted. Therefore, the TPR and FPR can not be interpreted directly as a good measure for the performance of the system, but rather a very rough estimate or indication of the performance. For the same reason, the TPR and FPR are presented in terms of number of trials in addition to the percentage.

A video of one of the successful attempts at the 1st sequence can be found here (youtu.be/ZvIhkh20hwU).

5.7.1 Performance of Subject 2 During Two-dimensional Online Drone Operation

For subject 2, the correct active class was detected in 19 of 27 cases during online operation of the drone, which gave a **TPR of 70.37** %. During the 27 times the subject intended to relax, there were no false positives, which gave a **FPR of 0.00 %**. This result was obtained from five attempts at sequence 1 and one attempt at sequence 3. A minimum of four commands were executed before the drone hit the barrier netting in all six attempts. In two attempts, all commands in the sequence were executed without the drone hitting the barrier netting. There were 15 attempts at forward movement, and 12 attempts at rotational movement. We note that the subject often had trouble with the rotation, i.e. right hand MI, with sensitivity for right hand MI being as low as 33.33 %, while sensitivity for left hand MI was 88.24 %.

In the online experiments without drone actuation, the subject did one attempt at performing sequence 1 two times consecutively without a break. This corresponded to 6 attempts of forward movement and 6 attempts of rotation. The correct active class was chosen in 11 of the 12 commands in the sequence, which gave a TPR of 91.17 %. There was one false positive, which gave a FPR of 8.33 %.

5.7.2 Performance of Subject 15 During Two-dimensional Online Drone Operation

During online operation of the drone by subject 15, the correct active class was detected in 32 of 34 cases, which means that a **TPR of 94.12%** was achieved. An active class was chosen 3 times in the 35 times the subject intended to relax, which gave a **FPR of 8.57 %**. This result was obtained from seven attempts at sequence 1 and one attempt at sequence 2. In all attempts, at least three commands were executed before the drone hit the barrier netting. In two attempts, all commands in the sequence were executed without the drone hitting the barrier netting. There was a total of 20 attempts of forward movement, i.e. left hand MI, and 14 attempts of rotation, i.e. right hand MI.

As for the online experiments without drone actuation, the subject did one attempt at performing sequence 1 two times consecutively without a break. This gave a total of 6 attempts of forward movement and 6 attempts of rotation. The correct active class was chosen in all of the 12 commands in the sequence, which gave a TPR of 100.00 %. There were no false positives, which gave a FPR of 0.00 %. While this is a very good result, we again emphasize that these numbers origin from a very small amount of data.

5.7.3 Three-dimensional Online Control

With the drone being inactive, both subjects got to try a short run of the three-dimensional control with three MI tasks. The results reflected the results with three MI tasks in section 5.6 - there was a high error rate. However, because only one short sequence of commands was tested for three-dimensional control due to limited time, the result is not specified, as it would be too uncertain to use for any conclusions.

5.8 The Impact of Artifacts

The fact that only frequency-domain and spatial filters were relied on for removal of oscillations not related to the ERD/ERS patterns in this work is worthy of critique. As artifact detection and removal was not implemented, it is necessary to investigate what impact this had on the results. Here, it is attempted to see whether artifact removal is something that could have improved the result by removing an element that is possibly confusing the classifiers, or if the artifacts correlate with the tasks and therefore make the classifiers rely on the pattern of the artifacts for classification. In the first case, the results in this work are good, but could have been even better if artifact removal was implemented. In the latter case, the results in this work could possibly look better than they are, as good classification results could be due to the classification of artifacts rather than actual brain signals.

This analysis is focused on EOG and EMG artifacts, as they are the main artifacts possibly impacting the frequency bands of interest, i.e. mu and beta bands, see section 2.2.2.

5.8.1 Effect of Mu-beta Band Pass Filter on Visible EOG Artifacts

First, it is interesting to see how an EOG artifact impacts the signal of other electrodes, and whether the artifact is visible when only considering the mu and beta bands. Therefore, a small portion of the data from subject 15, channels C3, C4 and Fpz, is plotted in fig. 5.20 and fig. 5.21.

In fig. 5.20, the signal is only notch filtered at 50 Hz and highpass filtered with a cutoff frequency at 0.1 Hz. Here, two EOG artifacts are clearly visible in Fpz at approximately sample no. 200 and 1000. One can also see that these two artifacts have propagated to channels C3 and C4, the most important channels for detection of right hand and left hand MI according to theory, because there is a small, momentary decrease in voltage at the same time as the EOG artifact in Fpz.

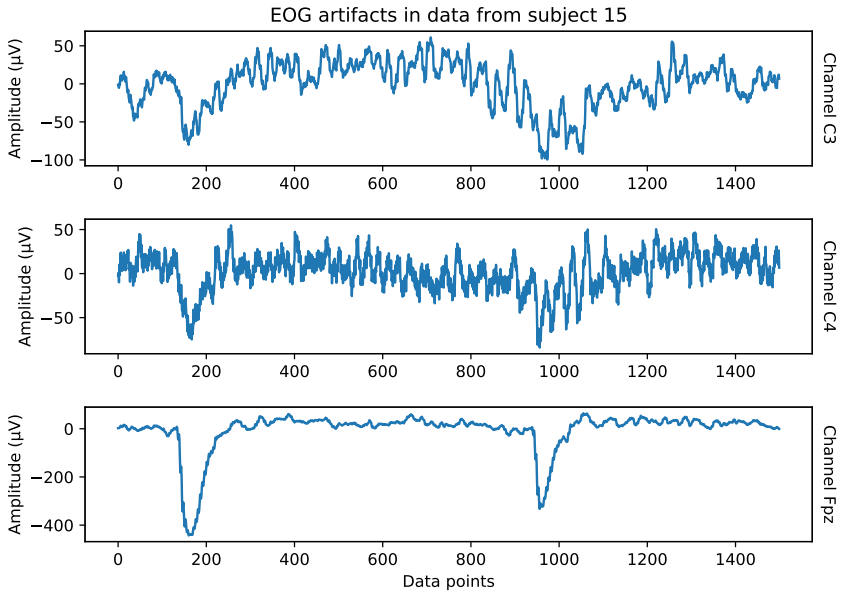


Figure 5.20: Two EOG artifacts showing in channels C3, C4 and Fpz in data of subject 15. The data is notch filtered at 50 Hz and highpass filtered with a cutoff frequency of 0.1 Hz.

The same signal, but bandpass filtered between 8 and 30 Hz to extract the mu and beta bands, is shown in fig. 5.21. The artifacts are still visible in channel Fpz as an abrupt and momentary increase in amplitude, but they are no longer clearly visible in channels C3 and C4.

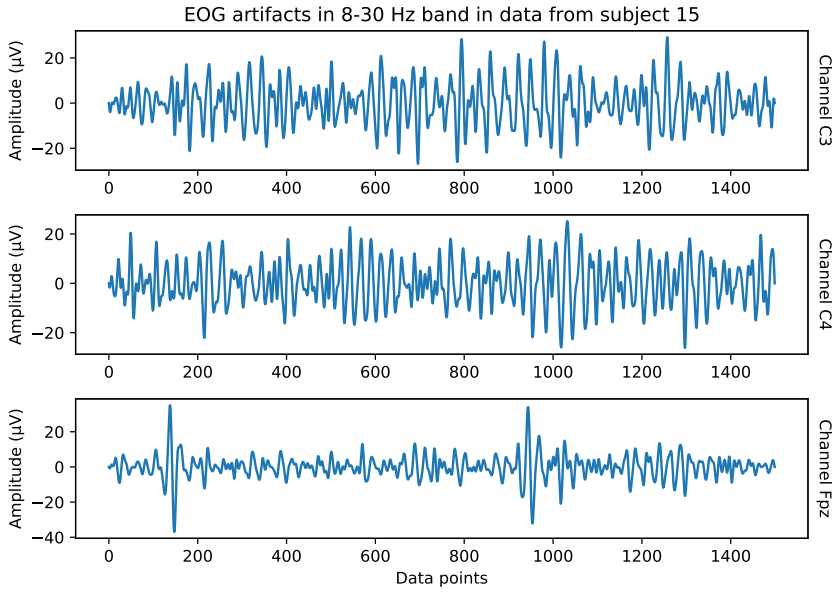


Figure 5.21: Two EOG artifacts showing in the Fpz channel in data of subject 15, when filtered bandpass filtered at 8-30 Hz. The artifacts are not clearly showing in channels C3 and C4.

The bandpass filter seemed to reduce effect of the artifacts considerably, as these artifacts did not overlap with the frequency bands of interest in the important channels. However, as this is just one example, this does not prove that the artifacts do not overlap with bands of interest in any cases. In other cases, the impact of the artifact in the mu and beta bands of important channels could be bigger, especially with EMG artifacts.

5.8.2 Correlation of Classes with EOG Artifacts

If the artifacts in any cases overlap with frequency bands of interest, the small possibility of the artifacts correlating with the classes is there. If there is a correlation between eye movements or eye blinks of the subject and the MI tasks, a reasonable belief is that the data of the subject could be somewhat correctly classified when only looking at channel Fpz. Due to its position, see fig. 5.22, eye movements are more apparent in Fpz than in other electrode locations.

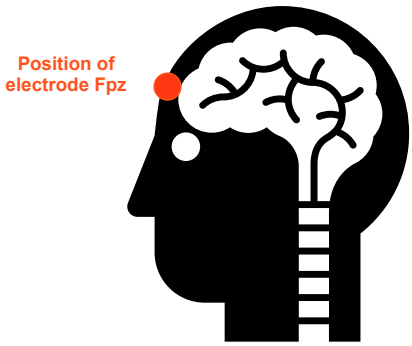


Figure 5.22: The position of the electrode Fpz on the head is marked with a red dot.

Therefore, if the accuracy is significantly higher than random, i.e. 33.33% with three classes, when classifying only based on the signal of this electrode, it is a clear warning sign that there could be a

correlation between the MI tasks and the eye movements of the subject. If this is the case, it could possibly mean that the eye movements improve the accuracy also when the Fpz electrode is not included in the analysis, as the EOG artifacts typically propagate to the other electrodes due to the connectivity of the scalp. Fpz is located well away from the motor cortex and will therefore not detect MI-related changes in the brain activity.

As the results of subjects 2 and 15 are most interesting, especially the online results in section 5.7, these are the most important results to investigate. Therefore, 10-fold cross-validation is performed on the accumulated data from sessions 1-4 with two MI tasks and resting state, for subject 2 and 15, using the best methods as obtained in section 5.4.7, with:

1. only the signal from the Fpz electrode, to see if there is any correlation at all between the eye movements and the classes. This result is denoted as *Fpz only* in the below table.

2. the signal from all electrodes *including* electrode Fpz, to see if this electrode can give an improvement from the earlier results, where Fpz has not been used. This result is denoted as *All channels incl. Fpz* in the below table.

3. the signal from all electrodes *except* electrode Fpz, i.e. electrodes CP1, CP2, FC1, FC2, C3, Cz, C4. This result is the same as in earlier experiments, and is provided again for comparison. It is denoted as *All channels excl. Fpz* in the below table.

The results are given in table table 5.27.

Table 5.27: Accuracy for subject 2 and 15 using only Fpz, all channels including Fpz or all channels exluding Fpz, for two MI classes and resting-state.

Subject	**Fpz only**	**All channels incl. Fpz**	**All channels excl. Fpz**
2	33.33 %	81.39 %	84.17 %
15	44.44 %	70.00 %	74.17 %

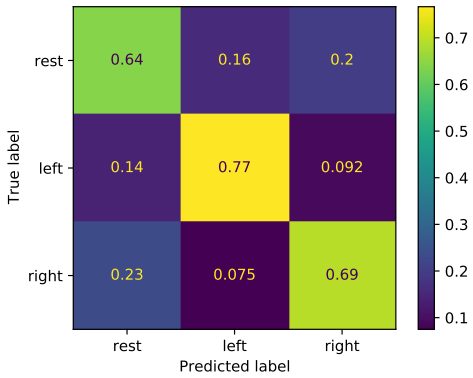
For subject 2, it seems that there is no possibility of deriving the correct classes from electrode Fpz, as the result when only using Fpz is random, and with all electrodes, the result decreases when including Fpz.

For subject 15, however, there seems to be some pattern in Fpz correlating with the classes, as the classification accuracy when only using channel Fpz is 11.11% above random. Nevertheless, the accuracy drops with 4.17 % from when Fpz is excluded to when it is included. To investigate this further, the confusion matrices corresponding to these experiments are plotted in fig. 5.23 and fig. 5.24.

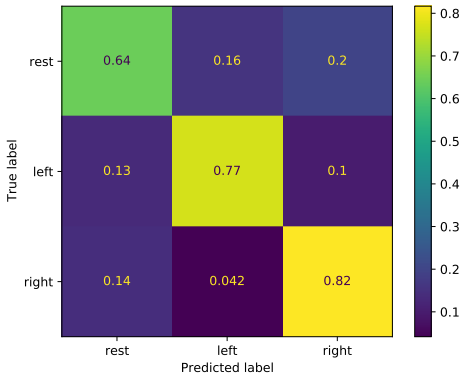
From fig. 5.23, it can be seen that the sensitivity of both rest and right hand imagery is quite high. However, the sensitivity of left hand imagery is close to zero. Left hand imagery is almost never predicted. It seems that by using only channel Fpz, it is to a high degree possible to distinguish rest and right hand imagery, but not left hand imagery.

Figure 5.23: Confusion matrix for classification of rest, right hand imagery and left hand imagery from data of subject 15, when only using the Fpz electrode.

From the confusion matrices in fig. 5.24, it can be seen that the sensitivity of rest and left hand imagery are equal with or without Fpz. The sensitivity of right hand imagery decreased with 13 % when Fpz was included. Right hand imagery is especially being confused with rest to a higher degree, but also with left hand imagery.



(a) Confusion matrix for classification with electrodes Fpz, C3, C4, Cz, FC1, FC2, CP1 and CP2.



(b) Confusion matrix for classification with electrodes C3, C4, Cz, FC1, FC2, CP1 and CP2, i.e. excluding Fpz. This is the same matrix as in section 5.4.6.

Figure 5.24: Confusion matrices for classification of resting-state, right hand imagery and left hand imagery from data of subject 15, with or without electrode Fpz.

The results when only classifying the signal from electrode Fpz could indicate for instance a correlation between eye movements and rest or MI, for example that the subject would blink only during rest. However, this is not reflected in the difference between classification with all electrodes except Fpz and all electrodes including Fpz. The fact that the accuracy decreases when Fpz is included suggests that Fpz only confuses the classifier. Hence, with subject 15, there *could* be a correlation of

EOG artifacts and the classes that could possibly help the classifier distinguish the rest state, but it seems likely that the artifacts primarily confuse the classifier. In subject 2, it seems very unlikely that EOG artifacts improve the result. Lastly, it is important to note that the possibility of EMG artifacts affecting the result was not ruled out.

Chapter 6

Discussion, Conclusion and Further Work

In this chapter, the acquired results and parts of the system are elucidated, in order to get a better understanding of the findings in this thesis and their weight. After the discussion, a conclusion is presented. Lastly, suggestions for further work are given.

6.1 Discussion

In this section, the results and the final system are evaluated and discussed. Weaknesses and strengths are identified.

6.1.1 Protocol and Dataset

The designed protocol can induce class-related eye movements, by leading the subject's eyes towards the arrow indicating a cue. Although the results in section 5.8 indicated that for subject 2 and 15, there were not any significant class-related eye movements that affected the results, this is not necessarily the case for the other subjects. Moreover, many of the subjects reported difficulties with not shifting the focus according to the cues, and these difficulties could disturb the execution of MI tasks. Therefore, the protocol could have been improved by displaying a cue or a fixation cross interchangeably, always at the middle of the screen, to keep the subject's focus at one point.

Additionally, subjects reported difficulties relaxing properly between tasks, as preparation or rehearsal of MI tasks could happen. Therefore, rest data could be contaminated by ERD/ERS induced from MI. This could possibly have been improved by having a randomized sequence of tasks. This would require some type of warning of the next task to be performed. Otherwise, the subjects' reaction time would either be long to perform the correct task, or they could start executing the wrong task due to stress.

Regarding artifacts, it was shown that it is unlikely that the classifiers rely on EOG artifacts for better classification results. However, EMG artifacts still exist, and were not analysed. As mentioned in section 2.2.2, they generally occur due to movements not correlated to cues. Therefore, it is concluded that it is unlikely that the results of subject 2 and 15 should be claimed invalid due to artifacts. However, it is emphasized that artifact detection and removal is a crucial part of BCIs, as artifacts can confuse the classifier and both decrease performance and be mistakenly interpreted as a source of

control. This is a weakness of this work.

Although data with both two and three MI tasks was recorded and evaluated for possible use online, it cannot be concluded that three MI tasks is too many to use with this system. This is due to different amounts of data being recorded per class for the two datasets. For dataset A, containing two MI classes, 120 trials of each MI class was recorded. For dataset B, only 40 trials were recorded for each MI class. Thus, there is not a proper basis for comparison between two and three MI tasks. The low performance with three MI tasks might be due to the small amount of data, rather than difficulty distinguishing three MI tasks. The system should therefore be tested with more data for three MI classes before concluding on the usability of that paradigm.

6.1.2 Results and Usability of the Optimization Algorithm

With NSGA-II, a handful of methods and parameters had to be chosen for analysis. Because the run-time of the algorithm increases with an increasing number of methods and parameters, it is not feasible to test all possible promising combinations. Therefore, some interesting methods and some tuning parameters were not included in the optimization problem. Notably FBCSP, which was not implemented for this work, and CNN, which was excluded due to the high run-time. Some parameters should have been tuned to give each methods a fair chance to perform their best, for instance the wavelet in DWT. Not to mention, other possible methods not discussed in this work.

When using NSGA-II, the results are highly dependent on the metrics used for the objectives. This is another reason that the conclusions the experiment are not necessarily the optimal solutions. Furthermore, it is a computationally expensive algorithm, so it may not be a suitable method for general use in fitting data processing pipelines to users in a complete BCI system. However, it was a useful tool in this thesis to compare methods and to choose a good method for each of the best-performing subjects.

A third reason the conclusions drawn from the NSGA-II experiment may be disputable, is that these conclusions are based on the way the pareto front is analysed, and this analysis is subjective. A simple approach was chosen: When analyzing the methods in the pareto front for subject 2 and 15, the method achieving the highest accuracy with all the accumulated data of the particular subject was chosen. However, there were typically several methods that were very close in accuracy to the chosen method. Therefore, it is also be possible to carry out a deeper analysis, taking into account different factors. One such factor that could be considered is the confusion matrix, especially the sensitivities for each class and the false negatives. It could be argued that in BCIs, one would rather want an overweight of false negatives during attempts at MI tasks, than false positives during intended rest. This could in some cases be satisfied by choosing one of the next-best methods, with an insignificant loss of accuracy.

6.1.3 Evaluating the System in Relation to the Ideal MI-based BCI

Ideally, the system would be compared to other similar, implemented BCIs. However, it is difficult to compare directly to other implemented systems due to the many possible variations in system design. For instance, number of MI tasks, mapping from MI task to command, epoch length and method of evaluation. As an example, it is difficult to compare this system to a system that was evaluated according to how accurately the user could make the drone follow a path. Moreover, many attempts in creating BCIs have stopped before online classification, due to the low accuracy on offline evaluation. Therefore, the system is evaluated in regards to an ideal BCI.

First, consider that a BCI should preferably work for as many subjects as possible. This complete system can only be concluded usable for two out of sixteen subjects, although it was not tested for other subjects than the two. Aiming for the system to work primarily with these two subjects was a choice made during the project. To make it usable for the other subjects, it would likely be necessary to record more data or train to enhance ERD/ERS patterns. Neither was possible with the amount of time available with the subjects. It was not surprising that only a few subjects mastered the BCI without training beforehand, as this is a well-known problem in MI-based BCIs.

The BCI system created in this thesis had a low throughput of commands. Due to the blocking state and long epochs, a command could effectively be sent only every four seconds. This could have been improved by using overlapping and/or shorter epochs. However, this would require parts of the system to be changed or redesigned, and an investigation of accuracy and reliability. Moreover, shorter epochs would also improve the response time. Yet, this would require low run time for the data processing and classification, which has not been assessed in this thesis. This should therefore be further investigated.

The low throughput and high response time also limits the usability of the system. Due to artifacts not being detected and removed, this system is only usable in a controlled setting where the subject is undisturbed and attempts to sit still.

With the state machine for control in two dimensions proposed in this work, the drone rotates or moves forward for a small, fixed amount of time. Therefore, the drone cannot be moved freely to all positions in the two-dimensional space, and the possibilities for control are very limited. However, it is easy to change the commands to activate more complex tasks consisting of a sequence of movements. As several earlier studies aimed at using a high number of MI tasks and did not obtain sufficient performance for online operation, this work aimed to implement a working BCI with a lower number of MI tasks.

6.2 Conclusion

The main goal of this project was to design and implement an asynchronous MI-based BCI for drone control. This was successfully carried out, using two MI tasks to control a drone in two dimensions. In regards to the objectives of this project, which were described in section 1.2, all objectives except the 3rd objective were satisfied. That is, sufficient and appropriate pre-processing was only partly implemented, as the crucial part of artifact detection and removal was missing.

A protocol for collection of MI EEG data was designed. 16 subjects were recruited and participated in recording of data with two MI tasks. 6 subjects were chosen for further analysis, and recording of data with three MI tasks.

Two out of 16 subjects, subject 2 and 15, performed vastly better than the rest for both two and three MI classes. By optimizing their classification pipeline using NSGA-II, an offline accuracy of 84.17% was achieved for subject 2 and 74.17% was achieved for subject 15, both with two MI classes and one rest class. Here, Riemannian-based feature extraction and CSP-based feature extraction were used respectively. For three MI classes and a rest class, subject 2 achieved an offline accuracy of 76.88% and subject 15 scored 67.50%, both using Riemannian-based features. These results were obtained with a flat structure of the classifier. A hierarchical structure was also proposed and tested, but was outperformed by the flat structure in all cases.

Through several stages of further testing, the classification scheme for three MI classes was found infeasible for online classification, possibly due to less data being recorded with three MI classes. For two MI classes, it was found feasible.

In a controlled environment, for short amounts of time, the two subjects controlled the drone in two dimensions with two MI tasks and rest state. The classifiers were trained with 4x8.5 minutes of EEG data, recorded approximately five weeks before online tests. For subject 2, Laplacian re-referencing and a 8-30 Hz bandpass filter was applied, before Covariance estimates were projected to the TS and classified with SVM. With subject 15, data was filtered to 8-30 Hz, before CSP and SVM was used.

During online operation of the drone, subject 2 achieved a TPR of 70.37 % and a FPR of 0.00 %, while subject 15 achieved a TPR 94.12 % and FPR of 8.57 %. Some online testing was also performed while the drone was inactive. In this case, subject 2 achieved a TPR of 91.17 % and a FPR of 8.33 %, while subject 15 achieved a TPR of 100.00 % and a FPR of 0.00 %. It is however emphasized that only a small number of short control sequences were performed, meaning these results are calculated from very small amounts of data. Thus, these results could serve as an indication of the system having a relatively good performance, but not as a valid basis for documenting the performance and comparing to other systems.

6.3 Further Work

As assessed throughout this thesis, there are several improvements that should be implemented for this work. First and most importantly, automatic artifact detection and removal should be implemented. A good candidate method is Independent Component Analysis (ICA), an algorithm widely used for removal of artifacts in MI-based BCIs [14, 15].

Furthermore, this system uses a very simple algorithm for avoiding duplicate commands when a single command was intended, the *blocking state*. This scheme seems effective at lowering FPR, but decreases the throughput of the system and can leave the system unoperable if a large amount of false positives are detected. Therefore, a more sophisticated algorithm for deriving control outputs from the classifier output should be investigated.

Regarding the low throughput, shorter time segments and/or overlapping segments could be investigated. Run-time of the data processing pipelines should be investigated to ensure low response time. Additionally, as only TPR and FPR were considered as metrics for evaluation of the system, additional metrics could be included in further work, such as response time and throughput.

FBCSP should be implemented and tested, as it has proved successful as an implementation for multi-class CSP, e.g. in several BCI competitions. This could improve accuracy from the implementation of multi-class CSP used in this work. In addition, CNN could be tested with larger quantum of data, as more data could improve deep learning methods. This could be done by having more extensive data acquisition of well performing subjects.

If the system is to be used with more subjects, an MI training scheme with feedback created based on data from high-performing subjects could be created, to enhance the performance of other subjects.

Moreover, it would be interesting to explore the possibility of implementing a subject-general BCI, that new subjects can use with no calibration of the classifiers. This could be done by training the classifiers with data from high-performing subjects. A different and interesting approach, is to explore the use of a CNN trained on data from the high-performing subjects, and then use transfer learning to calibrate the CNN with new subjects. Note that the new subjects will still need to be able to evoke the ERD/ERS patterns for this to work.

Lastly, more data should be acquired for three MI classes, to properly assess the feasibility of a MI-based BCI for three-dimensional drone control using the proposed approaches.

References

[1] Kjersti Brynestad. EEG-based motor imagery classification using dwt-based feature extraction with svm and riemannian geometry-based classifiers. Technical report, NTNU, 2020. DOI: 10.13140/RG.2.2.23210.13761.

[2] Erlend Vatsvåg. EEG-recorded motor imagery classification using emd-based features with svm and cnn classifiers. Technical report, NTNU, 2020. DOI: 10.13140/RG.2.2.36631.91043.

[3] Jonathan R Wolpaw, Niels Birbaumer, William J Heetderks, Dennis J McFarland, P Hunter Peckham, Gerwin Schalk, Emanuel Donchin, Louis A Quatrano, Charles J Robinson, Theresa M Vaughan, et al. Brain-computer interface technology: a review of the first international meeting. *IEEE transactions on rehabilitation engineering*, 8(2):164–173, 2000.

[4] Bernhard Graimann, Brendan Allison, and Gert Pfurtscheller. Brain–computer interfaces: A gentle introduction. In *Brain-computer interfaces*, pages 1–27. Springer, 2009.

[5] Natasha Padfield, Jaime Zabalza, Huimin Zhao, Valentin Masero, and Jinchang Ren. EEG-based brain-computer interfaces using motor-imagery: Techniques and challenges. *Sensors*, 19(6):1423, 2019.

[6] Niels Birbaumer and Paul Sauseng. Brain–computer interface in neurorehabilitation. In *Brain-Computer Interfaces*, pages 155–169. Springer, 2009.

[7] Chang-Hee Han, Klaus-Robert Müller, and han-jeong Hwang. Brain-switches for asynchronous brain–computer interfaces: A systematic review. *Electronics*, 9:422, 03 2020.

[8] Gernot R. Müller-Putz, Reinhold Scherer, Gert Pfurtscheller, and Rüdiger Rupp. Brain-computer interfaces for control of neuroprostheses: from synchronous to asynchronous mode of operation / brain-computer interfaces zur steuerung von neuroprothesen: von der synchronen zur asynchronen funktionsweise. 51(2):57–63, 2006.

[9] Dale Purves, George J Augustine, David Fitzpatrick, William C Hall, Anthony-Samuel LaMantia, James O McNamara, and S Mark Williams. Neuroscience, 2004.

[10] Qiong Gui, Maria V Ruiz-Blondet, Sarah Laszlo, and Zhanpeng Jin. A survey on brain biometrics. *ACM Computing Surveys (CSUR)*, 51(6):1–38, 2019.

[11] Andrea Biasiucci, Benedetta Franceschiello, and Micah M Murray. Electroencephalography. *Current Biology*, 29(3):R80–R85, 2019.

[12] Sten Grillner. Fundamentals of motor systems. In *Fundamental neuroscience*, pages 663–676. Elsevier, 2008.

[13] Weiwei Peng. EEG preprocessing and denoising. In Li Hu and Zhiguo Zhang, editors, *EEG Signal Processing and Feature Extraction*, pages 71–87. Springer Singapore, 2019.

[14] Mehrdad Fatourechi, Ali Bashashati, Rabab K. Ward, and Gary E. Birch. Emg and eog artifacts in brain computer interface systems: A survey. *Clinical Neurophysiology*, 118(3):480–494, 2007.

[15] Jose Urigüen and Begoña Zapirain. Eeg artifact removal – state-of-the-art and guidelines. *Journal of neural engineering*, 12:031001, 04 2015.

[16] Martin Lotze and Ulrike Halsband. Motor imagery. *Journal of Physiology-paris*, 99(4-6):386–395, 2006.

[17] Gert Pfurtscheller, Clemens Brunner, Robert Leeb, Reinhold Scherer, Gernot R Müller-Putz, and Christa Neuper. The graz brain-computer interface. In *Brain-Computer Interfaces*, pages 79–96. Springer, 2009.

[18] Kai J Miller and Jeffrey G Ojemann. A simple, spectral-change based, electrocorticographic brain–computer interface. In *Brain-Computer Interfaces*, pages 241–258. Springer, 2009.

[19] Benjamin Blankertz, Michael Tangermann, Carmen Vidaurre, Thorsten Dickhaus, Claudia Sannelli, Florin Popescu, Siamac Fazli, Márton Danóczy, Gabriel Curio, and Klaus-Robert Müller. Detecting mental states by machine learning techniques: the berlin brain–computer interface. In *Brain-computer interfaces*, pages 113–135. Springer, 2009.

[20] Jonathan R Wolpaw, Niels Birbaumer, Dennis J McFarland, Gert Pfurtscheller, and Theresa M Vaughan. Brain–computer interfaces for communication and control. *Clinical neurophysiology*, 113(6):767–791, 2002.

[21] Thorsten O Zander and Christian Kothe. Towards passive brain–computer interfaces: applying brain–computer interface technology to human–machine systems in general. *Journal of neural engineering*, 8(2):025005, 2011.

[22] Francisco Velasco-Álvarez and Ricardo Ron-Angevin. Asynchronous brain-computer interface to navigate in virtual environments using one motor imagery. In *International Work-Conference on Artificial Neural Networks*, pages 698–705. Springer, 2009.

[23] R Leeb, C Brunner, G Müller-Putz, A Schlögl, and G Pfurtscheller. Bci competition 2008–graz data set b. *Graz University of Technology, Austria*, pages 1–6, 2008.

[24] Clemens Brunner, Robert Leeb, Gernot Müller-Putz, Alois Schlögl, and G Pfurtscheller. Bci competition 2008–graz data set a. *Institute for Knowledge Discovery (Laboratory of Brain-Computer Interfaces), Graz University of Technology*, 16:1–6, 2008.

[25] Ronald Gordon and Edward J. Rzempoluck. Introduction to laplacian montages. *American Journal of Electroneurodiagnostic Technology*, 44(2):98–102, 2004.

[26] Syam, Syahrull Hi-Fi, Lakany, Heba, Ahmad, R.B., and Conway, Bernard A. Comparing common average referencing to laplacian referencing in detecting imagination and intention of movement for brain computer interface. *MATEC Web Conf.*, 140:01028, 2017.

[27] Shohei Tsuchimoto, Shuka Shibusawa, Seitaro Iwama, Masaaki Hayashi, Kohei Okuyama, Nobuaki Mizuguchi, Kenji Kato, and Junichi Ushiba. Use of common average reference and large-laplacian spatial-filters enhances eeg signal-to-noise ratios in intrinsic sensorimotor activity. *Journal of Neuroscience Methods*, 353:109089, 2021.

[28] Dennis J McFarland, Lynn M McCane, Stephen V David, and Jonathan R Wolpaw. Spatial filter selection for EEG-based communication. *Electroencephalography and clinical Neurophysiology*, 103(3):386–394, 1997.

[29] A.C. Fischer-Cripps. 3.5 - digital signal processing. In A.C. Fischer-Cripps, editor, *Newnes Interfacing Companion*, pages 269 – 283. Newnes, Oxford, 2002.

[30] Amara Graps. An introduction to wavelets. *IEEE Computational Science and Engineering*, 2(2):50–61, 1995.

[31] Emmanuel Didiot, Irina Illina, Dominique Fohr, and Odile Mella. A wavelet-based parameterization for speech/music discrimination. *Computer Speech & Language*, 24(2):341–357, 2010.

[32] Luis Alfredo Moctezuma and Marta Molinas. Classification of low-density eeg for epileptic seizures by energy and fractal features based on emd. *Journal of Biomedical Research*, 07 2019.

[33] Firas Jabloun and A Enis Cetin. The teager energy based feature parameters for robust speech recognition in car noise. In *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No. 99CH36258)*, volume 1, pages 273–276. IEEE, 1999.

[34] Kai Keng Ang, Zheng Yang Chin, Haihong Zhang, and Cuntai Guan. Filter bank common spatial pattern (fbcsp) in brain-computer interface. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 2390–2397. IEEE, 2008.

[35] Nicklas Stubkjær Holm, Sadasivan Puthusserypady, et al. An improved five class mi based bci scheme for drone control using filter bank csp. In *2019 7th International Winter Conference on Brain-Computer Interface (BCI)*, pages 1–6. IEEE, 2019.

[36] Moritz Grosse-Wentrup and Martin Buss. Multiclass common spatial patterns and information theoretic feature extraction. *IEEE transactions on Biomedical Engineering*, 55(8):1991–2000, 2008.

[37] Kai Keng Ang, Zheng Yang Chin, Chuanchu Wang, Cuntai Guan, and Haihong Zhang. Filter bank common spatial pattern algorithm on bci competition iv datasets 2a and 2b. *Frontiers in neuroscience*, 6:39, 2012.

[38] Fabien Lotte, Laurent Bougrain, Andrzej Cichocki, Maureen Clerc, Marco Congedo, Alain Rakotomamonjy, and Florian Yger. A review of classification algorithms for EEG-based brain-computer interfaces: a 10 year update. *Journal of neural engineering*, 15(3):031005, 2018.

[39] Sergios Theodoridis. Chapter 1 - introduction. In *Machine Learning*, pages 1 – 8. Academic Press, Oxford, 2015.

[40] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

[41] William S Noble. What is a support vector machine? *Nature biotechnology*, 24(12):1565–1567, 2006.

[42] Vojislav Kecman. Support vector machines–an introduction. In *Support vector machines: theory and applications*, pages 1–47. Springer, 2005.

[43] Suresh Balakrishnama and Aravind Ganapathiraju. Linear discriminant analysis-a brief tutorial. *Institute for Signal and information Processing*, 18(1998):1–8, 1998.

[44] Petros Xanthopoulos, Panos M Pardalos, and Theodore B Trafalis. Linear discriminant analysis. In *Robust data mining*, pages 27–33. Springer, 2013.

[45] David G Kleinbaum and Mitchel Klein. Introduction to logistic regression. In *Logistic regression*, pages 1–39. Springer, 2010.

[46] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[47] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)*, pages 1–6. Ieee, 2017.

[48] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.

[49] Dingjun Yu, Hanli Wang, Peiqiu Chen, and Zhihua Wei. Mixed pooling for convolutional neural networks. In *International conference on rough sets and knowledge technology*, pages 364–375. Springer, 2014.

[50] Sagar Sharma and Simone Sharma. Activation functions in neural networks. *Towards Data Science*, 6(12):310–316, 2017.

[51] Alexandre Barachant, Stephane Bonnet, Marco Congedo, and Christian Jutten. Riemannian geometry applied to bci classification. pages 629–636, Sep 2010. hal-00602700.

[52] Alexandre Barachant, Stéphane Bonnet, Marco Congedo, and Christian Jutten. Multiclass Brain-Computer Interface Classification by Riemannian Geometry. *IEEE Transactions on Biomedical Engineering*, 59(4):920–928, March 2012.

[53] Alexandre Barachant, Stéphane Bonnet, Marco Congedo, and Christian Jutten. Classification of covariance matrices using a riemannian-based kernel for bci applications. *Neurocomputing*, 112:172 – 178, 2013. Advances in artificial neural networks, machine learning, and computational intelligence.

[54] Olivier Ledoit and Michael Wolf. A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, 88(2):365–411, 2004.

[55] Alexandre Barachant, Stphane Bonnet, Marco Congedo, and Christian Jutten. Common spatial pattern revisited by riemannian geometry. pages 472–476, 2010.

[56] Xinjie Yu and Mitsuo Gen. *Introduction to evolutionary algorithms*. Springer Science & Business Media, 2010.

[57] Tinkle Chugh, Karthik Sindhya, Jussi Hakanen, and Kaisa Miettinen. A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms. *Soft Computing*, 23(9):3137–3166, 2019.

[58] Kalyanmoy Deb and J Sundar. Reference point based multi-objective optimization using evolutionary algorithms. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 635–642, 2006.

[59] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

[60] Luis Alfredo Moctezuma and Marta Molinas. Eeg channel-selection method for epileptic-seizure classification based on multi-objective optimization. *Frontiers in neuroscience*, 14:593, 2020.

[61] Luis Alfredo Moctezuma and Marta Molinas. Towards a minimal eeg channel array for a biometric system using resting-state and a genetic algorithm for channel selection. *Scientific RepoRtS*, 10(1):1–14, 2020.

[62] Benjamin Blankertz, K-R Muller, Dean J Krusienski, Gerwin Schalk, Jonathan R Wolpaw, Alois Schlogl, Gert Pfurtscheller, Jd R Millan, Michael Schroder, and Niels Birbaumer. The bci competition iii: Validating alternative approaches to actual bci problems. *IEEE transactions on neural systems and rehabilitation engineering*, 14(2):153–159, 2006.

[63] Michael Tangermann, Klaus-Robert Müller, Ad Aertsen, Niels Birbaumer, Christoph Braun, Clemens Brunner, Robert Leeb, Carsten Mehring, Kai J Miller, Gernot Mueller-Putz, et al. Review of the bci competition iv. *Frontiers in neuroscience*, 6:55, 2012.

[64] Pramod Gaur, Ram Bilas Pachori, Hui Wang, and Girijesh Prasad. A multi-class eeg-based bci classification using multivariate empirical mode decomposition based filtering and riemannian geometry. *Expert Systems with Applications*, 95:201–211, 2018.

[65] Li Zhang, Dezhong Wen, Changsheng Li, and Rui Zhu. Ensemble classifier based on optimized extreme learning machine for motor imagery classification. *Journal of neural engineering*, 17(2):026004, 2020.

[66] Robin Tibor Schirrmeister, Jost Tobias Springenberg, Lukas Dominique Josef Fiederer, Martin Glasstetter, Katharina Eggensperger, Michael Tangermann, Frank Hutter, Wolfram Burgard, and

Tonio Ball. Deep learning with convolutional neural networks for EEG decoding and visualization. *Human brain mapping*, 38(11):5391–5420, 2017.

[67] Yiran Zhao, Shuochao Yao, Shaohan Hu, Shiyu Chang, Raghu Ganti, Mudhakar Srivatsa, Shen Li, and Tarek Abdelzaher. On the improvement of classifying EEG recordings using neural networks. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 1709–1711. IEEE, 2017.

[68] Jasmin Kevric and Abdulhamit Subasi. Comparison of signal decomposition methods in classification of eeg signals for motor-imagery bci system. *Biomedical Signal Processing and Control*, 31:398–406, 2017.

[69] Karl LaFleur, Kaitlin Cassady, Alexander Doud, Kaleb Shades, Eitan Rogin, and Bin He. Quadcopter control in three-dimensional space using a noninvasive motor imagery-based brain–computer interface. *Journal of neural engineering*, 10(4):046003, 2013.

[70] Somsukla Maiti, Atanendu Sekhar Mandal, Santanu Chaudhury, et al. Classification of motor imagery eeg signal for navigation of brain controlled drones. In *International Conference on Intelligent Human Computer Interaction*, pages 3–12. Springer, 2019.

[71] Cédric Gouy-Pailler, Jérémie Mattout, Marco Congedo, and Christian Jutten. Uncued brain-computer interfaces: a variational hidden markov model of mental state dynamics. In *17th European Symposium on Artificial Neural Networks (ESANN 2009)*, page 77, 2009.

[72] Xu Duan, Songyun Xie, Xinzhou Xie, Ya Meng, and Zhao Xu. Quadcopter flight control using a non-invasive multi-modal brain computer interface. *Frontiers in neurorobotics*, 13:23, 2019.

[73] Jie Mei, Minpeng Xu, Lijie Wang, Yufeng Ke, Yijun Wang, Tzyy-Ping Jung, and Dong Ming. Using ssvep-bci to continuous control a quadcopter with 4-dof motions. In *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 4745–4748. IEEE, 2020.

[74] Ji-Hoon Jeong, Dae-Hyeok Lee, Hyung-Ju Ahn, and Seong-Whan Lee. Towards brain-computer interfaces for drone swarm control. In *2020 8th International Winter Conference on Brain-Computer Interface (BCI)*, pages 1–4. IEEE, 2020.

[75] Dae-Hyeok Lee, Ji-Hoon Jeong, Hyung-Ju Ahn, and Seong-Whan Lee. Design of an eeg-based drone swarm control system using endogenous bci paradigms. In *2021 9th International Winter Conference on Brain-Computer Interface (BCI)*, pages 1–5. IEEE, 2021.

[76] Bonkon Koo, Hwan-Gon Lee, Yunjun Nam, Hyohyeong Kang, Chin Su Koh, Hyung-Cheul Shin, and Seungjin Choi. A hybrid nirs-eeg system for self-paced brain computer interface with online motor imagery. *Journal of neuroscience methods*, 244:26–32, 2015.

[77] J. Philipp de Graaff. The PS-Drone-API. 2014.

[78] Magnus Själander, Magnus Jahre, Gunnar Tufte, and Nico Reissmann. EPIC: An energy-efficient, high-performance GPGPU computing research infrastructure, 2019.

[79] Sarah E Williams, Jennifer Cumming, Nikos Ntoumanis, Sanna M Nordin-Bates, Richard Ramsey, and Craig Hall. Further validation and development of the movement imagery questionnaire. *Journal of sport and exercise psychology*, 34(5):621–646, 2012.

[80] Ben Kröse, Ben Krose, Patrick van der Smagt, and Patrick Smagt. An introduction to neural networks. 1993.

Kjersti Brynestad and Erlend Vatsvåg

NTNU
Norwegian University of
Science and Technology