On Calibration Algorithms for Real-Time Brain-Computer Interfaces

Heskebeck, Frida

2023

Document Version:
Publisher's PDF, also known as Version of record

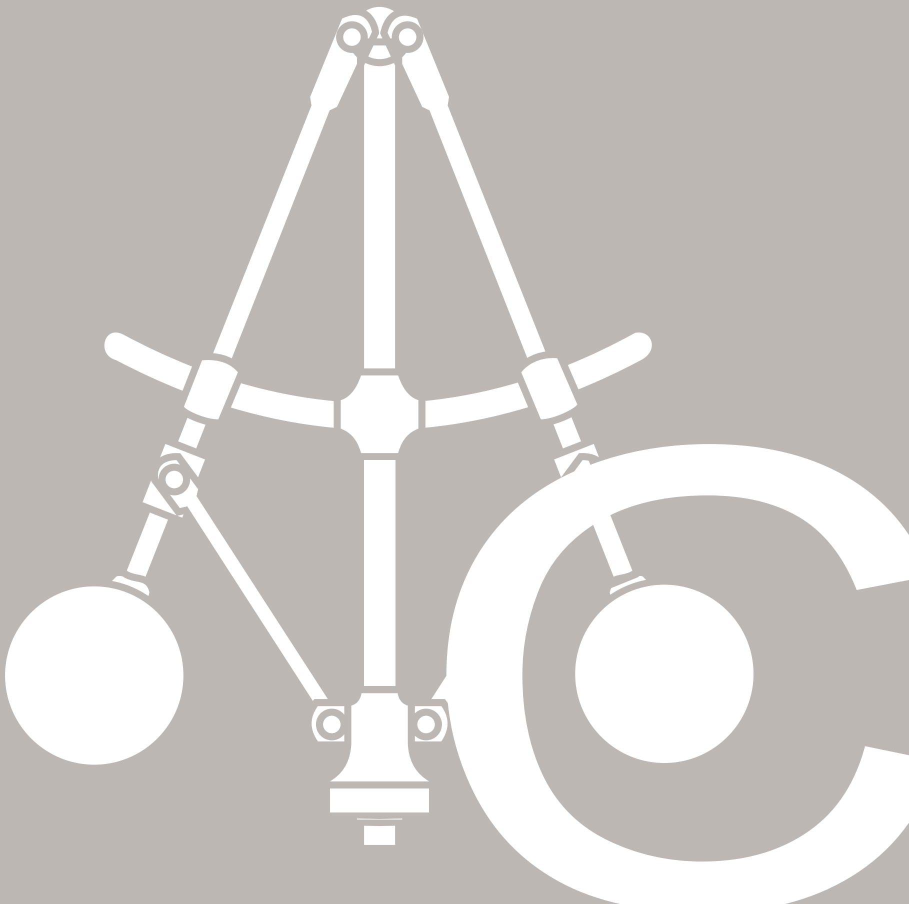
Link to publication

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

On Calibration Algorithms for Real-Time Brain-Computer Interfaces

FRIDA HESKEBECK
DEPARTMENT OF AUTOMATIC CONTROL | LUND UNIVERSITY



On Calibration Algorithms for Real-Time Brain-Computer Interfaces

Frida Heskebeck

LUND UNIVERSITY

Department of Automatic Control

Figure credits are given in a separate chapter at the end of the thesis.

Licentiate Thesis TFRT-3281
ISSN 0280–5316

Department of Automatic Control
Lund University
Box 118
SE-221 00 LUND
Sweden

Printed in Sweden by Media-Tryck.
Lund 2023

Abstract

A Brain-Computer Interface (BCI) is a system that, in real-time, translates the user's brain activity into commands that can be used to control applications, such as moving a cursor on the screen. The translation is made possible by machine learning methods and other algorithms. The thesis focuses on EEG-based BCIs which are the most common type of BCIs due to EEG measurements being non-invasive, having good temporal resolution, and being suitable for many applications. As of today, one of the biggest challenges for BCIs is the so-called calibration, which is necessary for the BCI to translate the user's brain activity correctly. The need for calibration comes from the variability of the brain signals over time and between users.

This thesis presents an extensive review of the state-of-the-art algorithms for BCIs, focusing on the calibration problem. Amongst the presented algorithms are methods for processing the EEG data, machine learning algorithms, and a brief introduction to transfer learning and Riemannian geometry. A more in-depth exploration of the so-called multi-armed bandits and Markov decision processes as possible methods to streamline the calibration procedure is presented, as well as a real-time framework for gathering and testing algorithms. Such a framework is crucial for testing new approaches for efficient calibration.

Acknowledgements

First, I want to thank my supervisors, Bo Bernhardsson and Carolina Bergeling, for all your support, inspiration, and help. I also want to thank my co-supervisor, Tore Hägglund, for your guidance in my Ph.D. studies.

Thank you!

Secondly, I want to thank the people at the Department of Automatic Control for many great moments at the “fika”. Special thanks go to Pex Tufvesson and Martin Gemborn Nilsson – for the fun times and good cooperation. Extra thanks to Pex for proofreading this thesis. To Emil Vladu – for being the best office mate ever and Eva Westin – for all administrative help during my parental leave.

Thank you!

Finally, I want to thank my family for all your support and love. Erik, Mom, Dad, and Freja. This would never be possible without you!

Thank you, I love you!

Financial support

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. The author is also a member of the ELLIIT Strategic Research Area.

Contents

1

Introduction

In a not-so-distant future, you are walking down the road with your hands full of things to carry. You receive a text message. Instead of picking up your phone, you ask your Brain-Computer Interface (BCI) system what the text says. But you don't ask out loud. You THINK, and your BCI understands. You answer the message by dictating a message in your head and reply without lifting a finger.

Across the street, you see a paralyzed person cruising down the road. She is controlling her wheelchair with her thoughts and seems to enjoy the freedom the BCI system has brought to her. You feel a bit nostalgic because you get a flashback to the day when your grandfather was able to communicate for the first time in thirty years, all thanks to a BCI system. Your BCI system recognizes your feeling and replays the memory for you. You remember your grandfather once telling you that memories used to only exist in one's head. Today, they are uploaded to the cloud and can be accessed from anywhere.

When you get home, you use your passthought to open the door. You see your crazy old neighbor a bit away. She is nice but refuses to use a BCI system. You two have had plenty of discussions about it and never managed to agree. She is convinced that the government can access everyone's thoughts and will soon be able to control you all. She says that "the first step towards destruction was when it was possible to download knowledge, who knows what other things will be inserted into your brains?". You, on the other hand, are convinced that the BCIs aid more people than they harm. Sure, there have been some accidents, especially in the beginning, but now? No, your BCI system is your assistant, and you trust it completely.

1.1 Brain-Computer Interfaces

Returning to the present world and this thesis you are reading, it is time to introduce Brain-Computer Interfaces (BCIs) a bit more. A BCI uses brain activity as input to the system, unlike today, where we typically use buttons, touchscreens, or our voice as input to a system. The 'system' can be anything from a cell phone to a car.

Brain activity is measured by some device from the head, often a so-called EEG device in the case of BCIs. A machine learning algorithm then decodes these brain signals. Every time before the BCI system is used, the machine learning algorithm must be trained to recognize the brain activity of that specific user. This is called *calibration* of the BCI system. Calibration usually takes a lot of time and is often very repetitive. My contributions in this thesis focus on methods to reduce the calibration time for EEG-based BCI systems.

In contrast to the futuristic world presented in the previous section, the use of BCI systems today is mainly medical and very limited, often aimed at improving the lives of persons with disabilities. There are many ethical concerns and technical hurdles to unravel before BCIs can become a part of our everyday lives.

1.2 Layout of thesis

The first chapters of this thesis give a wide-ranging background to BCIs. You will learn what a BCI is, what type of brain signals are typically used in BCIs, and what algorithms are used in BCIs. After that, I present my initial research on BCIs. I have investigated Markov Decision Processes and Multi-Armed Bandits as possible tools to improve the calibration of BCIs. Moreover, my colleagues and I developed a framework for creating BCIs that can be used to evaluate different algorithms for BCIs and study how the user behaves in a "closed-loop" system. The red thread through my work is the so-called *calibration* of BCIs, but more on that later. If you are already familiar with BCIs you can skip to chapter Chapter 6 where I start to present my research on BCIs. The bullet list below gives a summary of each chapter.

- Chapter 2: **Overview of BCIs** – This chapter gives background on BCIs, presents what types of BCIs exist, the history of BCI, and applications for BCIs.
- Chapter 3: **Background – Brain signals** – This chapter provides basic information about the brain, what brain signals are typically used for BCIs, and how these signals are measured. I will focus on EEG-based BCIs throughout this thesis since they are the most common.
- Chapter 4: **Background – Algorithms for BCIs** – This chapter presents state-of-the-art methods for decoding brain signals. It includes preprocessing of EEG data, methods for feature extraction, and machine learning algorithms. It also gives an introduction to Riemannian Geometry and transfer learning.
- Chapter 5: **Ethics** – This short chapter raises common ethical concerns related to BCIs.

- Chapter 6: **Problem formulation** – This chapter reviews the challenges with the calibration of BCIs and motivates why the calibration of BCIs is an interesting topic to study.
- Chapter 7: **Contributions** – This chapter lists my publications. I summarize the paper and state my contributions to each paper.
- Chapter 8: **Discussion** – This chapter discusses my current work and my ideas for future work.
- Chapter 9: **Final words** – Some final concluding words for the thesis.

References and Figure copyright

The background chapters include a lot of excellent references. Each section within the chapters ends with a "Section summary and References" paragraph where you find the references for that section. I recommend you look up the references for further information if you want to know more. The reference list with the full references is at page 76. The thesis includes a lot of figures, the source and copyright of all figures are found in the chapter Figure credits (page 74).

Contributions

My contributions presented in this thesis are twofold – first, the wide-ranging background presented in Chapters 2 to 5 and secondly, my published papers presented at the end of the thesis. The former contribution is meant to accommodate the need for a comprehensive yet simple-to-grasp introduction to the field of BCIs for anyone new to the field and interested in pursuing BCI-related research, particularly in the domain of calibration of BCIs.

2

Overview of BCIs

A Brain-Computer Interface (BCI) appears to be a bit like telekinesis in sci-fi movies: the user can control an application merely with their thoughts. In reality, BCIs use a lot of data processing and machine learning to decipher the meaning of the brain activity. Many challenges with BCIs limit their use (and make them far from being telekinesis). One of the significant obstacles is the so-called calibration of the BCI system, which I will introduce in this chapter. We will repeatedly touch upon the calibration of BCIs in the thesis but will take a closer look at it in Chapter 6.

In this chapter, I provide fundamental background on Brain-Computer Interfaces (BCIs). Section 2.1 introduces the procedure of a BCI, from collecting data to using the system. The section also defines the differences between different types of BCIs. Section 2.2 gives a short summary of the history of BCIs and shows the state-of-the-art applications. Finally, Section 6.1 returns to the calibration problem and gives a hint of my research.

2.1 What is a Brain-Computer Interface?

As introduced in Section 1.1, a Brain-Computer Interface (BCI) measures the brain signals and uses these as input to a system. The system can be anything from moving a cursor on a screen or a prosthetic arm to a communication system to workload detection for surgeons. There are many ways to measure brain activity and I will discuss these in Chapter 3. Throughout this thesis, I will focus on EEG-based BCIs. The measured brain signals need to be processed and a machine learning algorithm is typically needed to decipher the meaning of the signals (see Chapter 4 for details). To better understand BCIs, it is now time to look at the overall flow of a BCI system.

Fundamentals of a BCI system

Running a BCI system has two main parts: calibration and usage (see Figure 2.1). Calibration includes collecting data and training the machine learning model. Usage means using the BCI to control the application.

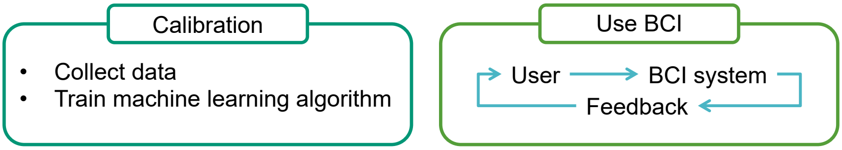


Figure 2.1 Illustration of the two main parts of running a BCI system. 1) A BCI system must be calibrated before it can be used. The calibration includes collecting data and training a machine learning algorithm. 2) Using the BCI system means that the user controls an application via the BCI system. The feedback to the user is what happened with the application, e.g., did the cursor on the screen move to the right when the user intended to move to the right? This is called human-in-the-loop.

In this section, the application of the BCI is moving the cursor on the screen. When the user imagines moving their right hand, the cursor should move to the right. Imagining left-hand movement moves the cursor to the left, imagining moving the tongue moves the cursor up, and finally, imagining moving the feet moves the cursor down.

Calibration. The *calibration* of a BCI system refers to the procedure of collecting data and training the machine learning model (also called machine learning algorithm). This might sound simple, but it is one of the biggest challenges with BCIs.

For a BCI system to interpret your intentions, the machine learning algorithm needs to be trained on labeled data, i.e., finding patterns in ground truth data where it is known what the user is thinking of. Hence, the first step of using a BCI system is to *collect data*. This is usually done with a stimuli program.

The stimuli program shows a prompt on a screen in front of the user, e.g., "right hand", and the user imagines moving their right hand for a few seconds until the next prompt is shown. The collected data then gets the label "right hand". We call each type of possible movement (right hand, left hand, tongue, feet) a class. The machine learning model needs multiple samples from each class to find patterns and classify the data correctly, thus understanding the brain activity. The process of showing stimuli to the user and then collecting data is repeated several times for all classes. Collecting data can take up to 20-30 minutes and is quite tiresome for the user. This is a massive problem for BCIs and one of the reasons we have no BCI applications in our everyday lives - the calibration takes too long time.

Once the labeled data is collected, the BCI system must learn how to decode the brain activity. A machine learning model is typically used for this purpose. The machine learning model needs to be adapted for this specific user which is called *training of the machine learning model*. Many different machine learning models exist (see Chapter 4 for details). Training a machine learning model takes time and computing resources but is usually faster than the data collection.

Now ask yourself this: Would you use a computer that needs 20 minutes of calibration to collect data and train the machine learning model before you can move the cursor on the screen? Probably not, which is why I in my research look into ways of reducing the calibration time for BCI systems (see Chapter 6).

Using a BCI. Once the BCI is calibrated, it is ready to use. In our example, the user can now imagine moving their right hand and the cursor will move to the right on the screen - which is pretty cool once it is up and running!

One can view this user-BCI interaction as a *feedback loop*. The user gives input to the BCI system by imagining movement. The BCI interprets the input and acts accordingly in other words, it decodes the brain activity and moves the cursor on the screen. Finally, the user observes what happens with the system and gives new input to the BCI. This feedback loop is closed by the human and is called human-in-the-loop. Such BCIs require fast processing of the data and cloud computing will likely be used to offload the heavy computations from mobile BCI devices in the future.

Types of BCI systems

Now that we know the basic flow of a BCI system, calibration and usage, it is time to distinguish between different types of BCIs.

BCI systems can be categorized based on four features: (i) the neuroimaging method, (ii) the BCI paradigm, (iii) passive vs. reactive vs. active BCI, and (iv) synchronous vs. asynchronous BCI. The categorization within these features is explained below.

(i) **Neuroimaging method** – Brain activity can be measured in different ways. The most common is electroencephalography (EEG), which I will focus on throughout this thesis. Some other possible measuring methods for BCIs are magnetoencephalography (MEG) and electrocorticography (ECoG). See Section 3.3 for more details and examples of measuring methods.

(ii) **BCI paradigm** – There are brain activity patterns that are typically used for BCIs. These patterns are called BCI paradigms. For example, if you are shown pictures of cats and dogs and are tasked to count the number of cats you see, your brain will react in a special way when you see a cat. This is called the P300 response and is one of the first used paradigms in BCIs. Another common paradigm is the motor imagery (MI) paradigm where the user imagines the movement of different body parts such as hands, feet, and tongue. See Section 3.2 for more details and examples of BCI paradigms.

(iii) **Passive vs. reactive vs. active BCI** – In a *passive BCI*, the user's thoughts are monitored but not acted upon by the BCI. It could be a BCI system where a surgeon's workload is measured during surgery. When the workload is high, the nurses should pay extra attention to aid the surgeon.

The user reacts to stimuli in a *reactive BCI*. An example is a so-called P300-speller, where letters are shown to the user, and when the letter the user wants to print is shown, the brain reacts in a recognizable way. Thus, the BCI can understand what letter the user wants to print.

In an *active BCI*, on the other hand, the user generates specific brain activity to control the BCI. An example is the control of a cursor on a screen through motor imagery movements. In such a system, the user imagines moving the right hand to move the cursor to the right.

(iv) **Synchronous vs. asynchronous BCI** – The final feature for BCIs is whether they are synchronous or asynchronous. In a *synchronous* BCI system, the user reacts to a cue or some kind of stimuli. It could, for example, be images shown on a screen. Thus, the brain and the stimuli are synchronized. The BCI still has to decode the meaning of the brain activity, but at least it knows when to look for it.

In an *asynchronous* BCI system, the user is free to give commands to the BCI at any time, as opposed to the synchronous BCI where the user only can give a command when the stimulus is given. An example of asynchronous BCI is where a computer cursor is controlled by the user imagining moving different body parts such as a hand. The BCI knows the paradigm but does not know when to look for the signal. For example, the BCI knows that it searches for imagined hand movements but not when the hands are imagined to move. An asynchronous BCI is more challenging to implement successfully than a synchronous BCI simply because it is not known when the interesting brain activity occurs.

Section summary and References

Summary. In this section, we have learned the fundamentals of Brain-Computer Interfaces (BCIs), highlighted the challenges with the calibration of BCIs, and distinguished between different types of BCIs. In the next section, we will briefly look at the history of BCIs and explore some applications of BCIs.

References. Nam et al. (2018) gives an excellent introduction to BCI systems while Nicolas-Alonso and Gomez-Gil (2012) gives a more in-depth explanation, both covering similar topics. The difference between passive and active BCI systems is reviewed in Krol et al. (2018). The problems around the calibration of BCI systems are discussed in Lotte (2015).

2.2 The History of BCI and Applications

We know how to calibrate and use a BCI from the previous section. In this section, we briefly examine the history of BCIs before we look at medical and entertainment-oriented applications of BCIs. Finally, we speculate about future BCI applications.

The applications mentioned here are not a complete list of applications but are some of the most commonly mentioned in the literature. I refer the interested reader to the references at the end of this section.

History

Hans Berger took the first steps toward Brain-Computer Interfaces in the 1930s when he invented a simple electroencephalogram (EEG) device. His device measured the brain waves, which opened the door to understanding how the brain works. In the 1980s, one of the first papers on BCIs was published. An early example of a BCI system was a communication system where the user is looking at the letter in a grid they want to print, a so-called P300-speller. Over the years, more ways of measuring brain signals and different BCI paradigms were discovered, opening doors for many BCI applications. Chapter 3 covers information about different measuring methods and BCI paradigms.

Medical applications

Historically, BCI applications have focused on the medical sector, aimed toward persons with disabilities, mainly because these persons would benefit the most from the technology. There are BCI systems for communication for locked-in patients, control of wheelchairs and prosthetic limbs, and rehabilitation. In a recurring competition called Cybathlon, contestants compete by completing everyday tasks using BCI systems. A task could be to plug in a light bulb with a prosthetic arm controlled by a BCI. The competition has the slogan "For a world without barriers".

The purpose of a medical application is typically to facilitate the user's life. It is generally acceptable with long calibration times as long as the final performance is flawless. Much effort is put into optimizing the system for the specific user in medical applications.

Entertainment-oriented applications

The difference between medical BCI applications and entertainment-oriented BCI applications lies in the purpose of the system. The purpose of medical BCI applications is to enhance or replace a function of the body, while the purpose of entertainment-oriented applications is simply "for fun". Examples of entertainment-oriented applications are: playing video games, creating art, playing chess, enhancing VR, controlling a toy such as a quadcopter, and meditation aids.

For entertainment-oriented systems, it is acceptable if the performance is not perfect and a fast calibration is often prioritized as opposed to in medical applications where performance is the highest priority. One of the reasons why we don't see BCIs in our everyday lives is the long calibration time, which removes the point of using such a system. Another reason is that the quality of the collected signals is usually bad. Common reasons for bad signal quality in an uncontrolled environment, such as your home, are the use of dry electrodes which are used for user convenience and are cheaper but have a worse connection than wet electrodes, and that there is more noise in an at-home setting. In a controlled environment on the other hand, such as a laboratory and medical application, the signal quality is good due to the noise-free environment, excellent quality of the electrodes, and professional supervision of the setup.

Futuristic applications

One can never know what the future brings, so we can only imagine. BCI is a promising interface between humans and technology, and many big companies are researching BCIs. One can imagine that we could use passthoughts rather than passwords to sign in to our accounts in the future. Another idea is that we could search the web for information without lifting a finger or perhaps even download knowledge in the future. It is possible that we could record dreams and replay them in the morning or even upload memories to the cloud. Artificial vision might be an alternative for blind persons and artificial hearing for people who are deaf or hard of hearing. The possibilities are endless, and more applications will arise with more research.

Section summary and References

Summary. In this section, we have looked into the past, current, and future of BCIs. Today, most applications are medical, but more entertainment-oriented applications are arising. Predicting the future is impossible, but many exciting applications are possible. In the next section, we will review the problem with calibration of BCIs.

References. Lotte et al. (2018b) gives an overview of the history of BCI. The paper by Berger (1929) is the first paper on EEG. The term Brain-Computer Interface was coined by Vidal (1973). Farwell and Donchin (1988) introduced the so called ”P300 speller”, one of the first BCI systems. See Lotte et al. (2018b) for more historical references.

Nicolas-Alonso and Gomez-Gil (2012) and Nam et al. (2018) give good reviews of applications for BCI systems. Nijboer and Broermann (2010) review the use of BCI system for Locked-in patients. *This Is CYBATHLON* (2022) is a competition in BCI systems. Muratore and Chichilnisky (2020) explore the possibility of artificial restoration of vision.

Aubé (2019) and Gonfalonieri (2020) give interesting ideas for future use of BCI systems in their blog posts. Ruiz-Blondet et al. (2016) write about using EEG devices for passthoughts. Neuralink (2021), Google (2021), Microsoft (2021), and Facebook (2021) are among other companies working with BCI research.

Chapter summary

In this chapter, we have learned the fundamentals of Brain-Computer Interfaces.

In Section 2.1, we saw that the calibration of BCIs is tedious for the user, though necessary for the BCI system to work. We talked about different types of BCIs which could be categorized based on what neuroimaging method is used, the BCI paradigm, whether it is active or passive, and if it is synchronous or asynchronous.

In Section 2.2, we briefly looked at the history of BCIs and explored different applications for BCIs.

The next chapter will teach us about how the brain works, BCI paradigms, and neuroimaging methods.

3

Background – Brain signals

From previous chapters, we now understand how Brain-Computer Interfaces (BCIs) work. Labeled data is collected and then used to train a machine learning algorithm in the calibration phase of a BCI system. Once the machine learning algorithm has found the patterns in the user's brain signals, the BCI system is ready for use. The next topic we will learn about is the brain and different types of brain activity.

Section 3.1 gives background on how the brain works, Section 3.2 explains different BCI paradigms, and Section 3.3 describes how different neuroimaging methods measure brain activity. Throughout the thesis, the main focus of neuroimaging methods lies on EEG. If you are familiar with this chapter's topics, you can skip to Chapter 4 which covers data processing and machine learning methods for BCIs.

3.1 Brain fundamentals

A BCI aims to use measured brain signals to control an application such as a prosthetic arm. To get some intuition for where these brain signals come from, we will first briefly examine the neurons in the brain and then different brain lobes.

Neurons

The brain and the nervous system control our bodies. The brain processes information from the rest of the body and responds with signals to move the muscles. Like the rest of the body, the brain is built of cells. The cells in the brain are called neurons (see Figure 3.1).

There are billions of neurons and they communicate with each other through electrical impulses. An impulse from one neuron to the next neuron is sent through the axon to the synapse which is the connection between neurons. The electrical impulse triggers the release of neurotransmitters in the synapse from the first neuron, the neurotransmitters then trigger the electrical signal in the second neuron, and thus, the signal has been transferred between the two neurons. Some substances affect the neurotransmitters or act as neurotransmitters. When the neurotransmitters

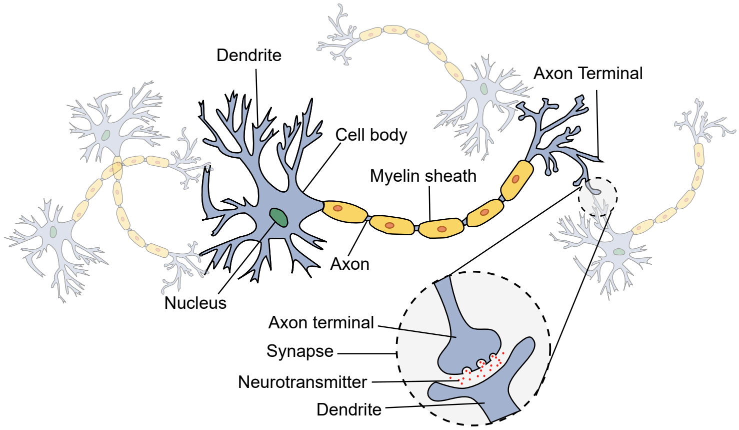


Figure 3.1 Schematics of neurons. The electrical impulse signal travels through the axon to the next neuron. A synapse connection is magnified, showing the release of neurotransmitters.

are affected, the brain activity is affected, ultimately affecting us in one way or another. One example of such a substance is caffeine in coffee which makes us more alert.

How the brain works is a vast and exciting topic, but the knowledge that brain activity originates from the neurons in the brain is enough for this thesis. I recommend the interested reader to dig into the references to find more information (though I want to warn you it is a fascinating topic, and you might be stuck reading about it for a long time).

Brain lobes

Billions of neurons form a complex network in the brain. It has been found that different parts of the brain, called brain lobes, process different information and thus have different functions, such as vision, motor control, and hearing. It is not entirely accurate to say that one part only does one thing and that one thing only happens in one part of the brain. But generally, the brain is said to be divided into different lobes. The following list presents the brain lobes and their functionalities as seen in Figure 3.2.

- **Prefrontal cortex** – The prefrontal cortex is located in the front of the head and is prominent in humans. This area is generally connected to decision-making, planning, and social skills.

- **Motor cortex and Sensory cortex** – At the top of the head, we find the motor cortex which controls the movement of our limbs. The motor cortex collaborates closely with the sensory cortex that processes bodily touch inputs. The sensory and motor cortex control different body parts in different areas of the

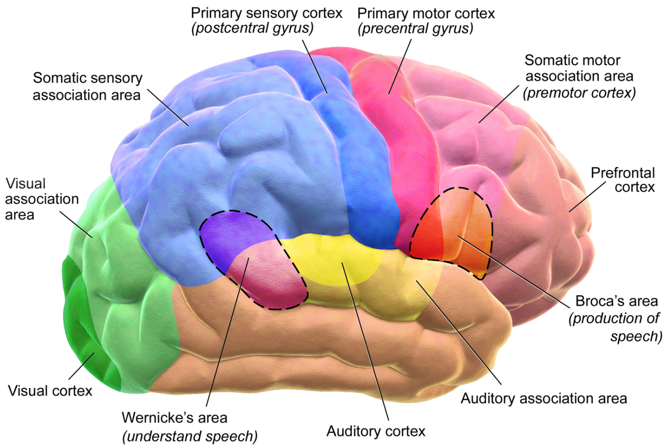


Figure 3.2 Schematics of the brain lobes. Each lobe generally processes a special type of brain signal, e.g., visual input in the visual cortex. The brain in the figure is viewed from the side, with the front of the brain to the right.

cortex (see Figure 3.3). Generally, the right half of the brain is connected to the left half of the body, and vice versa.

- **Visual cortex** – The visual cortex is located at the back of the head. Here, visual input is decoded. The information from the visual cortex propagates through the brain for further processing.

- **Auditory cortex** – The auditory cortex is located at the side of the head close to the ears. Here, sound is processed. Close by are also areas for speech understanding and speech production.

There are differences between the left and right hemispheres (halves of the brain). The motor cortex's right hemisphere controls the body's left half and vice versa. Generally, the left hemisphere handles language processing, problem-solving, and math. While the right hemisphere takes care of music, art, and empathy. However, it should be emphasized that one process is rarely isolated to one part of the brain, and these generalizations are very rough and should be taken with a grain of salt.

Section summary and References

Summary. In this section, we have learned the fundamentals of the brain's functionality. We have seen both how the neurons transfer information and where in the brain different information generally is processed. In the next section, we will learn about different BCI paradigms.

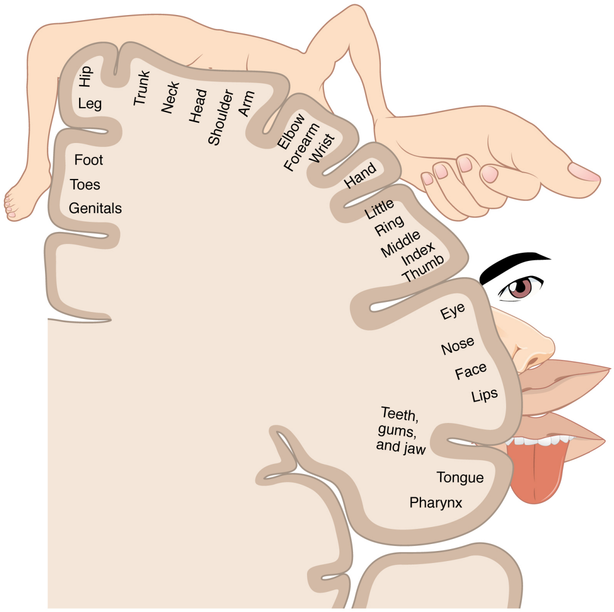


Figure 3.3 Scematics of which body parts are controlled where in the sensory and motor cortex. The left half of the brain is shown, viewed from the front.

References. See Purves et al. (2018) for a detailed description of the brain and its functionality. National Institute of Neurological Disorders and Stroke (2023) gives a general description of the brain's structure. King and Wyart (2021) study how multiple images are processed in the brain simultaneously.

3.2 BCI paradigms

Now that we have some intuition for how the brain works and where different kinds of information are processed in the brain, it is not too farfetched to realize that depending on what you are thinking, your brain activity will differ. For example, your brain will behave differently if you imagine moving your hand compared to if you look at images of cats and dogs.

There are specific types of brain activity that are often used in BCI systems. These are called BCI paradigms. Some common BCI paradigms are event-related potentials, motor imagery, steady-state evoked potentials, and mental workload. They are all explained in further detail below.

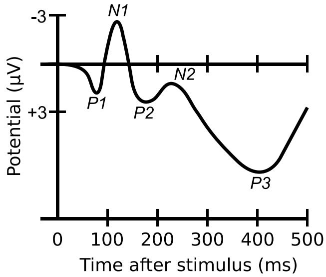


Figure 3.4 Plot of typical event-related potential components of a brain signal after stimuli onset.

Event related potentials (ERP) and P300

A common BCI paradigm is the event-related potentials, particularly the P300 paradigm. It is based on the fact that our brains react in a predictive way to things that happen around us, for example, if we see an image. A typical BCI setting for this paradigm is that a user is given stimuli (e.g., shown a sequence of images) and the brain activity in reaction to these stimuli is recorded. The moment the user is exposed to the stimuli is called stimuli onset.

The brain activity approximately 100 ms after the stimuli onset correlates to whether the stimuli were expected. This is called the N1 wave. The brain activity around 300 ms after stimuli correlates to if the stimuli were something to pay attention to, called the P300 response. Brain activity after 400-600 ms correlate to language understanding (see Figure 3.4).

P300. The brain activity after 300 ms is called the P300 response and is one of the most used BCI paradigms. One example is the P300-speller which is a communication system. All letters in the alphabet are arranged as a grid, and the user focuses on the letter they want to print. The rows and columns of the grid are lit up, and when the target letter is lightened, a P300 signal is detectable. The rows and columns are lit up multiple times so that the P300 signal can be averaged from many trials. This P300-speller was one of the first BCI setups in the early days of BCIs.

An advantage of the P300 paradigm is that the P300 signal is a spontaneous reaction in the brain to stimuli. Hence, all users can use a P300-based BCI system. Another advantage is that the P300 signal is easily identified from averaged data. A drawback is that a lot of data is required to get the average, which means showing the stimuli multiple times for the user. Another disadvantage is that the BCI system is limited to the user reacting to stimuli rather than the user deciding when to give an input.

Error Potentials (ErrP). Error-related potentials are another subcategory of event-related potentials. Error potentials arise when something ’wrong’ happens, e.g., the user tries to move the cursor on a screen to the right but the cursor moves to the left. Error potentials have been used to correct BCI systems' actions.

Motor Imagery (MI)

A second common BCI paradigm is the Motor Imagery (MI) paradigm. In the motor imagery BCI paradigm, the user imagines moving their limbs. It is typically hands, feet, or tongue. Different limbs are controlled in different areas of the brain (as we saw in the previous section, Section 3.1), so depending on what area of the brain is active the BCI system can decode what movement was imagined. In a motor imagery BCI system, the movement of different limbs is connected to different commands such as "imagining lifting the left hand" means moving the cursor on the screen to the left.

An advantage of the motor imagery paradigm is that the BCI system can be designed so that the user decides when to give the command, unlike some other BCI paradigms where the user can only react to given stimuli. One problem with this paradigm is that not everyone can successfully imagine the movement of limbs and these users can't use a motor imagery-based system at all. For users who can imagine movements, much calibration is needed for the BCI system to work.

Steady-state evoked Potentials (SSxEP)

A third common BCI paradigm is the steady-state evoked potentials. The steady-state evoked potentials are generated by flickering stimuli, for example a flickering light. The frequency of the brain activity synchronizes with the frequency of the flickering stimuli. If the stimuli are visual, it is called Steady-State Visual Evoked Potentials (SSVEP). There are also steady-state auditory evoked potentials (SSAEP) and steady-state somatosensory evoked potentials (SSSEP).

In an SSVEP-based BCI, multiple sources of flickering lights are shown, each connected to a command (e.g., the slow flickering light corresponds to the input 'no' and the fast flickering light corresponds to 'yes'). The user then focuses on the flickering light corresponding to the command they want to give. The BCI then identifies the frequency in the brain activity and can thus find the command the user aimed for.

As with P300, most persons can use BCI systems based on SSVEP and the signals are relatively easy to identify. The biggest obstacle is that it is tiresome for the user to concentrate on flickering stimuli for a longer period.

Mental workload

The mental workload is a final, slightly less common type of BCI paradigm. The workload increases if there are many tasks to focus on simultaneously or if it is a very hard task. The aim of a BCI system based on mental workload could be simply to monitor the workload or aid the user in some way to decrease the workload. An example would be to monitor the workload of a surgeon during surgery. If the workload is high, the nurses need to aid the surgeon.

An advantage of workload-based BCIs is that all users can use them. A drawback is that the user can't choose an action in the same sense as the other paradigms. Thus, workload detection gives a more passive BCI system.

Section summary and References

Summary. This section explored four BCI paradigms: P300 (event-related potentials), motor imagery, SSVEP, and workload. The workload paradigm is the least used of these four. All paradigms have advantages and disadvantages and allow for different kinds of BCI systems and applications. The need to calibrate the BCI system before it can be used persists for all paradigms and remains as one of the biggest challenges with BCIs. In the next section, we will learn about neuroimaging techniques.

References. Abiri et al. (2019) give a thorough review of the different BCI paradigms. Sur and Sinha (2009) give a short review of the typical ERP components of brain signals. Pfurtscheller and Neuper (2010) give a more detailed review on the MI paradigm, Riggins and Scott (2020) on the ERP paradigm, and Vialatte et al. (2010) on the SSVEP paradigm. Finally, Krol et al. (2018) discuss BCI systems based on mental state detection.

3.3 Neuroimaging methods

Now that we know what type of brain activity is interesting for BCIs, the next question is how this brain activity is measured.

There are many neuroimaging techniques (ways to measure brain activity). In this section, I describe some that can be used in BCI systems. The neuroimaging methods can be categorized based on the features below. See Figure 3.5 for a graphical comparison between the neuroimaging techniques.

- **Temporal resolution** – Temporal resolution means how precise in time the measurements are like how long time between samples.
- **Spatial resolution** – Spatial resolution means how precise in space the measurements are like the distance in mm or cm between sample locations.
- **Invasive/Noninvasive** – An invasive neuroimaging method requires surgery or injection of some chemical compound or electronics into the body. A noninvasive method measures brain activity from outside of the body.
- **Direct/Indirect** – As discussed in Section 3.1, the brain's neurons transmit information through electrical impulses. This is called electrophysiological activity and is what a direct neuroimaging method measures.

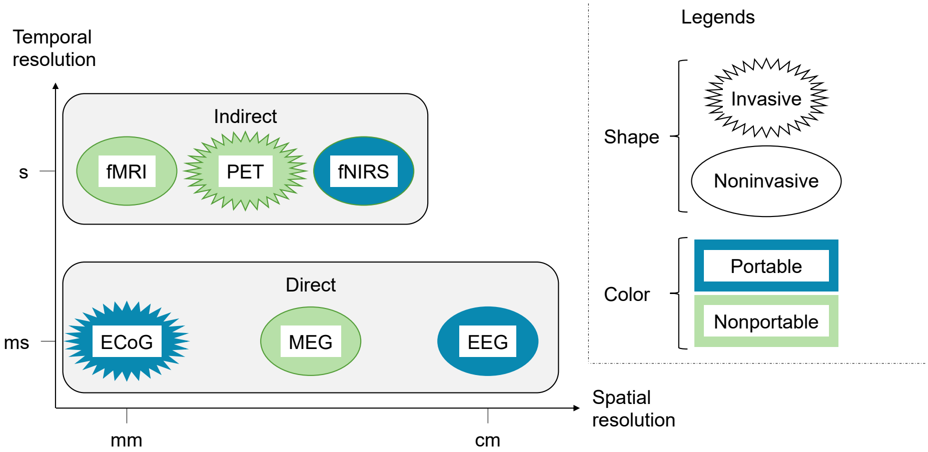


Figure 3.5 Comparision of different neuroimaging techniques: positron emission tomography (PET), functional magnetic resonance imaging (fMRI), functional near-infrared spectroscopy (fNIRS), electrocorticography (ECoG), magnetoencephalography (MEG), and electroencephalography (EEG). Five features of the neuroimaging methods are illustrated: temporal resolution is shown at the y-axis, spatial resolution at the x-axis, invasive/noninvasive is shown by the shape of the nodes, portable/nonportable by the color of the nodes, and direct/indirect is shown by the grouped areas. (Figure inspired by Nam et al. (2018)).

When the brain is active, it consumes more energy and oxygen, which increases blood flow to the active part of the brain. This is called the hemodynamic response of the brain and is what an indirect neuroimaging method measures.

- **Portable/Nonportable** – A portable neuroimaging device can be used while moving. A nonportable device requires the user to be still. This is typically a result of the size of the equipment. Nonportable neuroimaging devices are often huge scanners in shielded rooms that cannot be moved, while portable devices often are caps the user wears on the head.

The most used neuroimaging method for BCI systems, which I focus on throughout this thesis, is electroencephalography (EEG) since it has a good temporal resolution, is relatively cheap, and is relatively user-friendly. Some other ways to measure the brain signals are magnetoencephalography (MEG), electrocorticography (ECoG), functional magnetic resonance imaging (fMRI), functional near-infrared spectroscopy (fNIRS), and positron emission tomography (PET). All these methods are described below, but my main focus is on EEG.

Table 3.1 Features of EEG.

Electroencephalography (EEG)	
Feature	**Description**
Temporal resolution	Milliseconds
Spatial resolution	Centimeters
Invasiveness	Noninvasive – electrodes on scalp
Measurement strategy	Direct – potential difference between electrodes
Portability	Portable

Electroencephalography (EEG)

The first neuroimaging method I will describe is electroencephalography (EEG) which is the most common for BCI systems. Table 3.1 gives an overview of the features of EEG and are explained in more detail in the following text.

EEG measures brain activity through the potential difference between electrodes resulting from the neurons' activity. Hence, EEG is an example of a direct neuroimaging method. The EEG electrodes are placed on the user's scalp, often attached to a cap to keep them in place (see Figure 3.6). Since the electrodes are placed on the outside of the head, it is a noninvasive neuroimaging method. The temporal resolution is in the range of milliseconds, which is good and comes from EEG being a direct neuroimaging method. It is preferable if the user is stationary during measurement to reduce artifacts from movement, but the user could also be free to move. Thus, EEG is considered a portable neuroimaging method. The equipment required to measure EEG signals is relatively simple, making EEG one of the cheapest neuroimaging alternatives. One drawback of EEG is the low spatial resolution, which lies in the range of centimeters. Since the EEG electrodes are placed on the user's scalp, it is mainly the brain activity close to the brain's surface that can be measured. Despite the low spatial resolution, EEG is the most popular neuroimaging method, probably due to being a relatively user-friendly method.

Since I focus on EEG-based BCIs, we will discuss EEG in more detail compared to the other neuroimaging methods. First, I will discuss how the EEG electrodes are placed on the scalp in the so-called 10-20 system, then some typical features of EEG signals, and finally, different types of EEG electrodes.

10-20 system. The EEG electrodes are arranged on the user's scalp in a standardized way called the "10-20 system" (see Figure 3.7). A minimum of three electrodes are required for EEG measurements: a ground electrode, a reference electrode, and a measuring electrode. Usually, more electrodes are used and are arranged according to the 10-20 system. Nowadays, it is common with 24, 32, 64, or 128 electrodes, but other variants exist.

The letters in the 10-20 system refer to the cortex lobe above which the electrode is placed (Fp - prefrontal, F - frontal, C - central, P - parietal, T - temporal, and O

Figure 3.6 EEG recording device. The electrodes are placed on a cap that the user wears on the head.

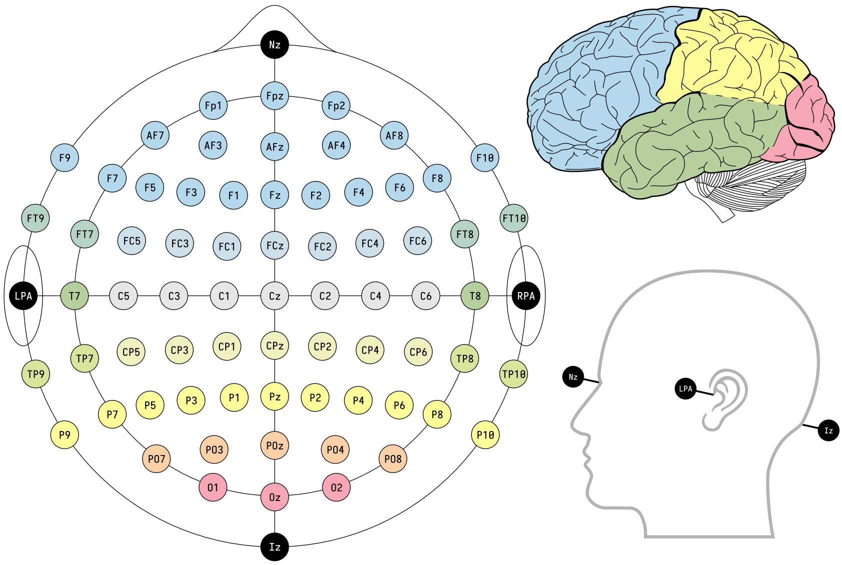


Figure 3.7 10-20 system. The left half of the figure shows the EEG electrode placement on the head seen from above. The nose is at the top and the ears are at both sides. The electrodes and the brain's colors show where the electrodes are placed. The head in the bottom right corner shows the nose, neck, and ear points used when placing the EEG electrodes on the head.

- occipital) and the numbers refer to the hemisphere (half of the brain). Electrodes with odd numbers are located on the left hemisphere, and even numbers above the right hemisphere. There is also a "z" which means that the electrode is placed between the hemispheres. The name "10-20 system" refers to the 10% or 20% distance between the electrodes compared to the "nose-to-neck" and "ear-to-ear" distance.

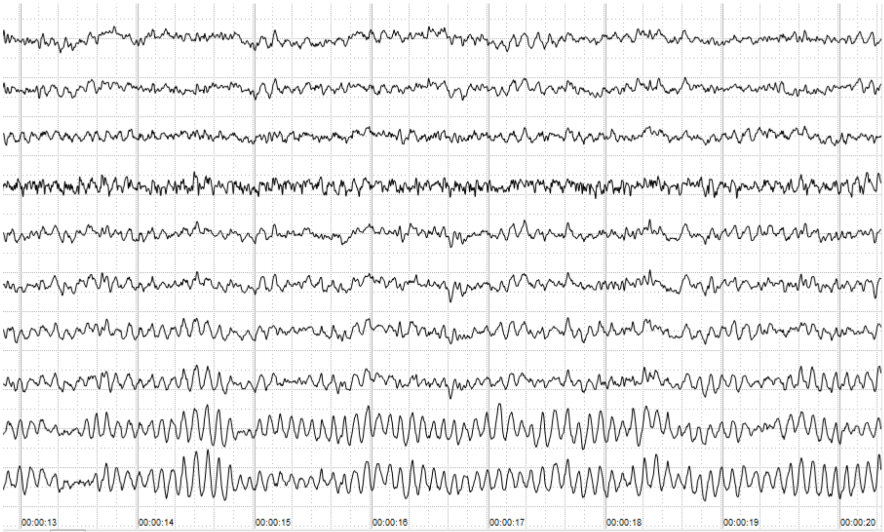


Figure 3.8 EEG signals for ten channels over seven seconds.

EEG signals. Each electrode measures the brain activity in the spatial vicinity of the electrode compared to the reference. It is nearly impossible to know exactly where in the brain the signal originated from as the spatial resolution is low. There is ongoing research on methods that aim to trace the origin of brain activity seen in EEG data to its source in the brain, so-called source reconstruction. Currently, source reconstruction is not a standard tool for BCIs, but it might be in the future. It is often enough for BCI systems to find patterns in the EEG data, but BCIs would arguably benefit from more detailed brain activity data.

Viewing EEG signals, the data from one electrode corresponds to one "line". The electrodes are often called channels and are vertically stacked to create a joint plot of the EEG signals (see Figure 3.8). The signals are often sampled at a frequency of 100-500 Hz, which gives a good temporal resolution of what is happening inside the brain.

Typical oscillating waves (brainwaves) can be seen in EEG signals. The brainwaves are divided into frequency bands: delta, theta, alpha, beta, and gamma (see Figure 3.9). The distinguishing characteristics between these bands are their frequency and amplitude. Frequency is how fast the signal oscillates and is measured in waves per second (Hz). It is possible to measure frequencies up to half of the sampling frequency which means that EEG sampled at 100 Hz can measure frequencies in the range 0-50 Hz. Amplitude is how big the signal is and is measured in microvolt (µV) and typically lies in the range 10-20 µV.

The different frequency bands arise depending on the current activity in the brain (see Table 3.2 for a summary). The *delta* (1-4 Hz) and *theta* waves (4-8 Hz) are typically observed in sleeping persons. *Alpha* waves (8-13 Hz) are seen when

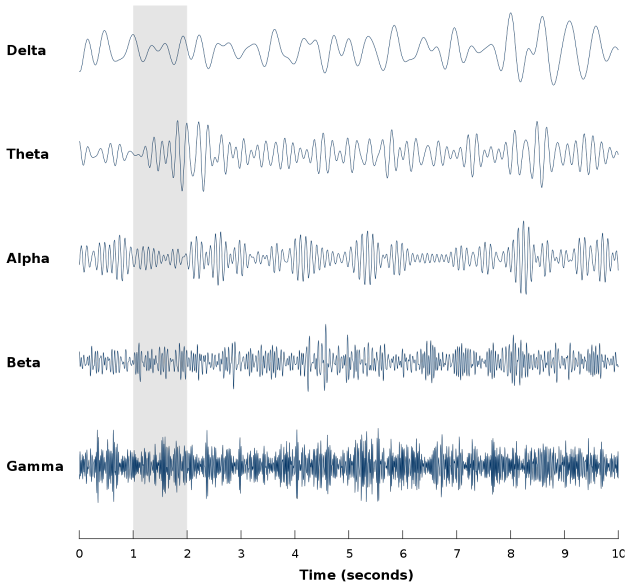


Figure 3.9 Oscillating delta, theta, alpha, beta, and gamma waves that are typically observed with EEG.

Table 3.2 Overview of frequency bands in EEG.

Power band	Frequency band (Hz)	Occurance
Delta	1-4	Sleep
Theta	4-8	Sleep
Alpha	8-13	Relaxation
Beta	13-30	Thinking
SMR	13-15	Muscular movement
Gamma	30-100	Problem solving

a person is relaxed, such as during meditation. *Beta* waves (13-30 Hz) occur when a person is thinking and keeping focus. There is a subband in the beta band, the *sensimotor rythm* (SMR) band (13-15 Hz), which is related to muscular movement. Finally, *gamma* waves (30-100 Hz) are observed when a person is solving a problem.

As with everything related to the brain, the reality is not black and white. These bands and during what activity they occur is a very generalized view. Nevertheless, this view of specific frequency bands is used when analyzing brain activity from EEG data.

Table 3.3 Features of MEG.

Magnetoencephalography (MEG)	
Feature	**Description**
Temporal resolution	Milliseconds
Spatial resolution	Millimeters - Centimeters
Invasiveness	Noninvasive – measurement outside of head
Measurement strategy	Direct – magnetic field around head
Portability	Nonportable

Wet vs dry electrodes. The EEG electrodes need to be in contact with the skin for them to measure brain activity. Electrodes are considered wet if they require a conductive gel to get the connection with the skin and dry if they have direct contact with the skin. The signal quality is generally better for wet electrodes than for dry ones since the wet electrodes have better contact with the skin.

Applying the gel takes some time and the gel needs to be washed away afterward, making the wet electrodes a bit more cumbersome than the dry electrodes to use. The dry electrodes are easy to apply but are designed as spikes to get through the hair, which can be uncomfortable for the user after a while. Some dry electrodes are flat but can (for most users) only be used on the forehead where there is no hair.

Whether wet or dry electrodes are best depends on the use case. Do you need an easy-to-apply EEG device with poor signal quality or a harder-to-apply device with better signal quality? Many new dry electrodes are developed aiming to be comfortable and give good signal quality, all to make BCI systems as user-friendly as possible.

Magnetoencephalography (MEG)

Even though EEG is the big star of the different neuroimaging techniques for BCIs, we will still discuss other methods. Next up is magnetoencephalography (MEG), whose features are summarized in Table 3.3.

MEG measures the magnetic field around the brain that arises from brain activity. It is a noninvasive and direct neuroimaging technique but requires advanced and expensive hardware that is nonportable (see Figure 3.10). MEG can measure brain activity in the range of milliseconds which is the same as EEG, but MEG has a spatial resolution in the range of millimeters to centimeters which is more precise than EEG. Despite the good spatial resolution, MEG is rarely used in BCI systems due to the expensive and nonportable equipment.

Electrocorticography (ECoG)

The next neuroimaging method is electrocorticography (ECoG) whose features are summarized in Table 3.4.

Figure 3.10 MEG recording device.

Table 3.4 Features of ECoG.

Electrocorticography (ECoG)	
Feature	**Description**
Temporal resolution	Milliseconds
Spatial resolution	Millimeters
Invasiveness	Invasive – surgery required to place electrodes
Measurement strategy	Direct – potential difference between electrodes
Portability	Portable

ECoG is a direct neuroimaging technique based on the same principles as EEG. The difference is that the ECoG electrodes are placed directly on the brain rather than on the scalp, as with EEG. Surgery is required to place the ECoG electrodes on the brain. Hence, it is an invasive neuroimaging method. Compared to EEG signals, the signals from ECoG measurements have a better spatial resolution, a wider frequency range, less sensitivity to noise and artifacts, and a higher amplitude. The temporal resolution of ECoG is in the range of ms, slightly better than EEG. Overall, the signal quality from ECoG is better than for EEG, which mainly results from the ECoG electrodes being placed directly on the brain while the EEG electrodes are placed on the scalp. As EEG, ECoG is a portable neuroimaging method. The main reason why ECoG is not widely used in BCIs is the need for surgery to place and remove the electrodes.

Functional magnetic resonance imaging (fMRI)

The next neuroimaging technique is functional magnetic resonance imaging (fMRI) whose features are summarized in Table 3.5.

fMRI is based on the indirect measurements of the hemodynamic response in the brain, i.e., the effects of changes in blood flow due to brain activity. When one

Table 3.5 Features of fMRI.

Functional magnetic resonance imaging (fMRI)	
Feature	**Description**
Temporal resolution	Seconds
Spatial resolution	Millimeters
Invasiveness	Noninvasive – big scanner outside head
Measurement strategy	Indirect – magnetic properties of blood
Portability	Nonportable

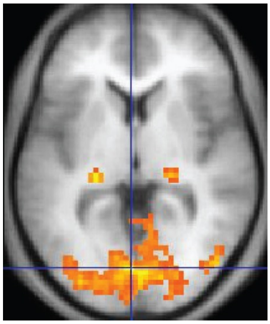


Figure 3.11 Image from fMRI scan. Shows a cross-section of the brain with the front of the head at the top of the image. The colored area shows the active regions in the brain.

part of the brain is active, more oxygenated blood flows to that part of the brain. The magnetic properties between blood with a lot of oxygen and little oxygen are different, which is what fMRI measures in a noninvasive way using a big scanner outside the head. The spatial resolution is in the millimeter range, similar to EEG (see Figure 3.11). On the other hand, the temporal resolution is slower than EEG, in the range of seconds. This is a result of the speed of the hemodynamic response. Blood can't flow infinitely fast, and the blood flow is always delayed compared to the brain activity. Heavy computations are required to analyze a scan, and before fast computers, it took hours or days to get the result from the fMRI scan. The equipment for fMRI is expensive and nonportable, making fMRI unsuitable for BCI systems.

Functional near-infrared spectroscopy (fNIRS)

The next neuroimaging method is functional near-infrared spectroscopy (fNIRS) whose features are summarized in Table 3.6.

fNIRS is, as fMRI, based on the indirect measurements of the hemodynamic response in the brain. fNIRS utilizes the different light-absorbing properties of blood with varying oxygen levels instead of the magnetic properties that fMRI uses. The fNRIS electrodes are noninvasive and placed outside the scalp (see Figure 3.12).

Table 3.6 Features of fNIRS.

Functional near-infrared spectroscopy (fNIRS)	
Feature	**Description**
Temporal resolution	Seconds
Spatial resolution	Millimeters to centimeters
Invasiveness	Noninvasive – cap outside head
Measurement strategy	Indirect – light absorbing properties in blood
Portability	Portable

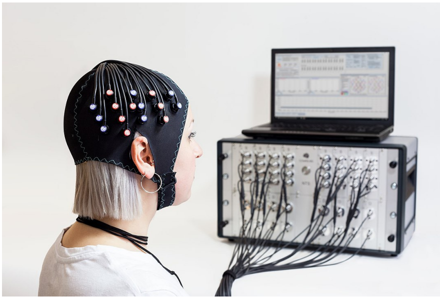


Figure 3.12 fNIRS cap and recording device.

The electrodes emit infrared light, and the blood's light absorption is measured. The oxygen level in the blood can be determined from the measured light absorption, and thereby, the brain's activity since active brain regions consume more oxygen. The light penetrates only a few centimeters into the brain but has a good spatial resolution for that area. The hemodynamic response limits the temporal resolution and is in the range of seconds.

Even though fNIRS measurements are sensitive to movement artifacts, it is considered a portable measurement method. It has potential for home applications in BCIs since it is inexpensive and relatively user-friendly.

Positron emission tomography (PET)

The last neuroimaging technique I will describe is positron emission tomography (PET) whose features are summarized in Table 3.6.

PET is based on positron emission from a tracer molecule introduced into the user's bloodstream. For example, the tracer molecule could be a glucose molecule variant, which is accumulated in brain regions with high activity. PET measurement relies on the hemodynamic response and is an indirect neuroimaging method. Since the method requires injecting the tracer molecule into the bloodstream, it is considered an invasive method. Other sources might argue that PET should be considered a noninvasive method since no surgery is needed. A big scanner is needed to mea-

Table 3.7 Features of PET.

Positron Emission Tomography (PET)	
Feature	**Description**
Temporal resolution	Seconds
Spatial resolution	Millimeters
Invasiveness	Invasive – injection of tracer molecule
Measurement strategy	Indirect – accumulation of tracer molecule
Portability	Nonportable

Figure 3.13 PET scanner.

sure the positron emission. Hence, the PET is nonportable (see Figure 3.13). The spatial resolution is in the range of mm, and the temporal resolution is in the range of seconds. PET is rarely used in BCIs due to advanced and expensive equipment and the low temporal resolution.

Section summary and References

Summary. In this section, we have learned about different neuroimaging methods. The main focus was on EEG, but MEG, ECoG, fMRI, fNIRS, and PET were also covered. The following section is a summary of this whole chapter.

References. Both Nicolas-Alonso and Gomez-Gil (2012) and Nam et al. (2018) include reviews of the methods for measuring brain activity. Marzbani et al. (2016) gives a good review on oscillating EEG signals. Müller-Putz (2020) briefly compares wet and dry electrodes.

Chapter summary

In this chapter, we have learned about the brain.

In Section 3.1, we learned that the brain is built out of neurons that communicate with electrical impulses and neurotransmitters. Then, we learned that the brain generally processes different types of brain signals in different parts of the brain.

In Section 3.2, we learned about the different BCI paradigms used in BCIs. The most common are the P300, motor imagery, and SSVEP paradigms.

In Section 3.3, we learned about different neuroimaging methods, particularly EEG. We learned how the EEG electrodes are arranged in the 10-20 system, the typical frequency bands observed in EEG data, and the difference between wet and dry electrodes.

In the next chapter, we will learn how to decode the meaning of the measured brain signals.

4

Background – Algorithms for BCIs

From Chapter 1 and Chapter 2 we know what a Brain-Computer Interface (BCI) is and that it needs to be calibrated to work. Calibration of a BCI means running a stimuli program to collect labeled data and then training a machine learning algorithm with the collected data. We remember that one of the biggest challenges for BCIs is that the calibration is tedious and that I therefore focus on improving the calibration in my research. From Chapter 3, we know that EEG is the main neuroimaging method for BCIs and we are familiar with the different BCI paradigms such as motor imagery and P300.

A BCI system uses a predetermined neuroimaging method (EEG in our case) and BCI paradigm. The BCI paradigm decides the classes the brain activity can belong to. For example, if the motor imagery paradigm is used, some possible classes are left hand, right hand, tongue, and feet. The user controls the BCI by imagining moving a body part. The classes in a P300 experiment are attended and not attended. The user controls the BCI by counting the number of times the object they are attending to is shown, e.g., a letter they want to print. See Section 3.2 for details on the different BCI paradigms.

Labeled data is collected during calibration (see Section 2.1 for details on how). Labeled data means knowing what the user was thinking about - the data has a label. For example, in a motor imagery-based BCI, the label specifies what body part the user imagined moving, e.g., the right hand. The labeled data is used during calibration for training the machine learning algorithm and is called training data. After calibration, when the BCI system is used, the new data has no label and the purpose of the BCI is to predict the class of the data. The predicted class is then used to control the application in some way, e.g., move the cursor on the screen to the right.

This chapter presents the procedure for decoding EEG data and the algorithms that are used for this. The first step to decode EEG data is to preprocess the raw continuous EEG data (Section 4.1). The second step is to find features in the pre-

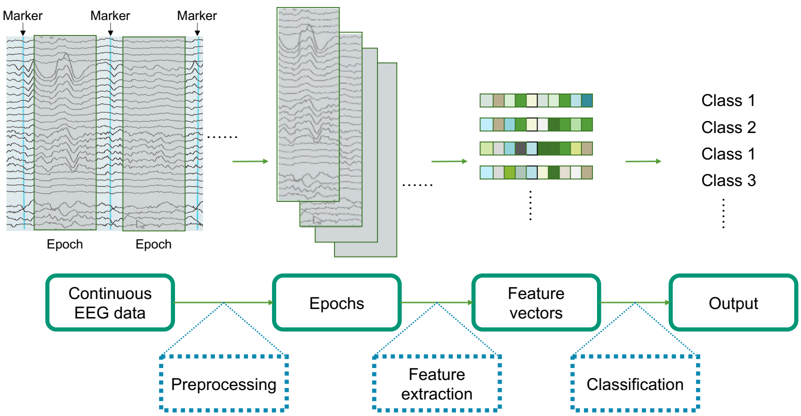


Figure 4.1 Flowchart of the steps to decode brain signals. The decoding starts with raw continuous EEG data that is divided into epochs in the preprocessing step. In the feature extraction step features for the epochs are derived and stored in feature vectors. The feature vectors are used for classification and the final output is the predicted class for the brain activity for each epoch.

processed data (Section 4.2). The final step is to classify the data with a machine learning algorithm (Section 4.3). Section 4.4 gives an introduction to Riemannian geometry and Section 4.5 presents some transfer learning methods for BCIs. Finally, Section 4.6 lists some useful Python tools for EEG-based BCIs.

There are a lot of terms that are used when talking about data related to BCIs. Table 4.1 defines some of these terms. Figure 4.1 illustrates the three steps to classify EEG data: preprocessing, feature extraction, and classification. There is no clear cut between what counts as preprocessing and what counts as feature extraction. In this thesis, I have made the distinction that preprocessing is what is done with the data until the data is split into epochs. Feature extraction is what is done with the data after epoching but before it is put into the machine learning algorithm, and classification is the classification of data with a machine learning algorithm. However, as you will discover, some preprocessing methods can be used for feature extraction and vice versa. I will do my best to clarify this confusion for the relevant methods described below but bear in mind that we are operating in a gray area.

4.1 Preprocessing

The first step to decode brain signals is to preprocess the data. Preprocessing is needed regardless of the used neuroimaging method. In this section, I present pre-

Table 4.1 Table defining some terms related to BCI data.

Term	**Description**
Continuous raw data	The continuous raw data is all EEG data from an experiment. The EEG data is annotated with markers, e.g., for stimuli onset.
Stimuli onset	Stimuli onset is the moment a stimulus is presented to the user. The stimuli can, e.g., be pictures for the user to react to or prompts (e.g., right hand) for motor imagery.
Marker	A marker is a timestamp in the EEG data. It often marks stimuli onset, but there can be markers for other events, such as the start or pause of the experiment.
Epoch	An epoch is a chunk of EEG data containing interesting brain activity.
Trial	One trial refers to one stimulus shown. While the term *epoch* generally refers to the actual EEG data, the *trial* refers to the experiment. For example, if it is a P300 experiment and the user is shown pictures and is tasked to count the number of cats, a trial is when the user is shown a picture and reacts to it. The corresponding epoch contains the EEG data from that trial.
Session	One session contains many trials that are done one after the other. Using the P300 picture experiment again, one session is the full sequence of trials where the user is shown picture after picture and reacts to them all. Multiple sessions with the same user can be performed in one day or on different days. EEG data from one session often refers to the collection of epochs from the session but could also refer to the continuous raw data from the session.
Subject/user	Subject refers to the person performing the experiment, i.e., the user. I choose to use the term *user* in this thesis instead of subject since the term subject can be interpreted as a bit dehumanizing.
Dataset	A dataset contains data from multiple sessions and users acquired using the exact same experimental setup. An excellent place to find public EEG datasets is MOABB [Jayaram and Barachant, 2018].

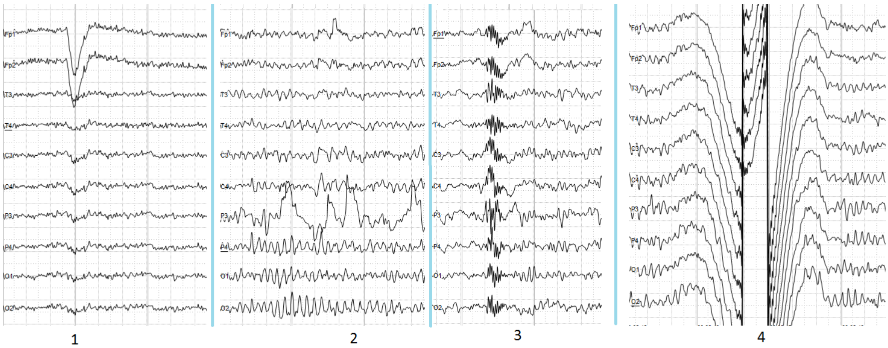


Figure 4.2 EEG data with artifacts. 1. Eye movements, e.g., blinking. Mainly visible in frontal electrodes. 2. Bad contact between one electrode and skin. 3. Artifact from swallowing. 4. Bad contact between the reference electrode and skin affects all channels.

processing methods for EEG data. Other preprocessing methods are needed if other neuroimaging methods are used.

The purpose of the preprocessing step is to clean the EEG data from artifacts and isolate the parts of the EEG data that contain the interesting brain activity.

Artifact removal

The first preprocess action is to remove disturbances in the EEG data. The disturbances are called artifacts and affect the EEG data but do not originate from brain activity (see Figure 4.2). Artifacts in EEG data can be grouped into two categories: physiological artifacts (1 and 3 in Figure 4.2) and technical artifacts (2 and 4 in Figure 4.2). Physiological artifacts originate from the user's body, such as muscular activity, eyeblinks, or heartbeats. Technical artifacts originate from all other sources than the user. One example is the power grid that injects 50 Hz noise into the recordings (60 Hz in e.g. the USA). Another technical artifact is electrodes with a bad contact to the skin.

Artifacts can be removed manually while examining the data. However, manually removing artifacts in a BCI system is impossible and they need to be removed automatically. One method to identify physiological artifacts is by adding extra sensors that keep track of the source, e.g., an electrode close to the eye to record eyeblinks or close to the heart to record heartbeats. For example, the technical artifact from the power grid can be removed with a notch filter. Careful electrode placement can prevent a bad connection between skin and electrodes.

One approach is to discard the EEG data containing artifacts altogether. The other approach is to separate the artifact from the interesting EEG data and only remove the artifact. Linear filtering, independent component analysis, or other methods can be used to remove the artifact while keeping the EEG data.

Independent Component Analysis - ICA. One algorithm to identify and remove artifacts while keeping the EEG data is the Independent Component Analysis (ICA). Other algorithms exist but I present ICA as one representative example. ICA is not specific to EEG signals and can be applied to other types of signals, e.g., sound.

The idea behind Independent Component Analysis is that the vector of data $\mathbf{x}(t)$ is a mixture of the vector of independent sources $\mathbf{s}(t)$ and noise $n(t)$ as

$$\mathbf{x}(t) = f(\mathbf{s}(t)) + n(t). \tag{4.1}$$

The mixing function f can be assumed to be either linear or nonlinear. ICA uses an algorithm (e.g., Infomax or FastICA) to find the mixing function. Then, the sources $\mathbf{s}(t)$ are extracted and can be used for further data analysis.

In the BCI setting, $\mathbf{x}(t)$ is the measured EEG signals, $\mathbf{s}(t)$ is the independent sources of brain activity or other sources, and $n(t)$ is random noise. Artifacts independent of the brain activity can be removed with ICA since they end up as individual source $\mathbf{s}(t)$. ICA could also be used as a feature extraction method to get independent sources of brain activity within the brain. As I warned before, what counts as a preprocessing and feature extraction method is a gray area, and ICA is an example of this.

ICA is limited to removing artifacts independent of the EEG signal, e.g., artifacts from eye movement. The assumption on the mixing function f (nonlinear or linear mixing function) is accompanied by limitations such as high complexity and hard-to-solve problems for the nonlinear case and sometimes a too simple model in the linear case.

Bandpass filter

Leaving artifact removal and continuing with the next preprocessing method, we are now at bandpass filtering.

Recalling Section 3.3, we remember that EEG signals oscillate at different frequencies depending on the ongoing brain activity. Bandpass filtering keeps only the components of EEG data with frequencies within the filtering range. This means that information outside the filtering range is attenuated. This requires knowledge about in what frequency range the interesting brain activity is found, i.e., what frequencies to keep. Through bandpass filtering, noise and uninteresting brain activity are removed.

Bandpass filtering is almost always included as a preprocessing step but could also be used as a feature extraction method in, for example, a mental workload-based BCI. A mental workload-based BCI aims to detect "how hard the brain is working". As we know from Section 3.3, the different frequency bands (alpha, delta, theta, and gamma) roughly correspond to different workloads. Thus, the interesting features of a mental workload-based BCI are the powers for all frequency bands, i.e., how much of each frequency band there is in the EEG data.

As for ICA, we see that bandpass filtering is an example of a preprocessing method that can be used both for preprocessing and feature extraction.

Re-referencing

The next thing to consider in the preprocessing step is the referencing of the EEG data. From Section 3.3 we know that EEG measures the potential difference between each electrode and the reference electrode. The reference electrode is preferably placed in an inactive zone, such as the earlobe or the base of the neck. However, brain activity could still spill over to the reference electrode, affecting the recorded data in all of the electrodes. One way to handle this is to re-reference the EEG data. Other reasons to re-reference the EEG data are to improve the signal quality and the signal-to-noise ratio (SNR). Noteworthy is that re-referencing is not a preprocessing method that is always done, in contrast to e.g., bandpass filtering which is a method that is often used.

Several methods exist to re-reference the EEG signals, such as Bipolar reference, Surface Laplacian Reference, and Common Average Reference (CAR). Here, I only mention these methods and refer the interested reader to the references for this section for more details. The general approach in all these methods is to combine other electrodes into a reference value to which the electrode at hand is compared.

Epoching

The final important part of the preprocessing step is to cut the data into epochs. An epoch is a chunk of data (often a few seconds) that is known to contain brain activity that is to be decoded. For example, in a P300 experiment, the user is shown a picture and we want to decode the user's reaction to the picture. An epoch would be the EEG data collected from when the picture was shown until a few seconds after. The *stimuli onset* (when the image is shown) is marked with a *marker* in the EEG data. The first part of Figure 4.1 shows a graphical illustration of epoching continuous raw EEG data.

Baseline correction is often applied to ensure that all epochs are comparable.

Baseline correction. Before the stimuli onset, there are usually a few seconds where the user is supposed to be in a blank state, meaning that they should not think about anything. The EEG data from the blank state can be used for baseline correction, which means adjusting the EEG data so all of this 'blank' data is at the same level. When the baseline for all epochs is at the same level, the epochs can be compared fairly.

Baseline correction is sometimes such an obvious part of epoching that it is not mentioned. However, baseline correction is not always done, so one has to pay close attention when reading papers or using public datasets.

Section summary and References

Summary. In this section, we have taken a closer look at the preprocessing step which is the first step to decode EEG data. Important parts of the preprocessing step are artifact removal, bandpass filtering, and epoching. In the next section, we will look at the second step to decode brain signals: feature extraction.

References. Artifact removal is discussed in Nicolas-Alonso and Gomez-Gil (2012) and Nam et al. (2018). Nicolas-Alonso and Gomez-Gil (2012) also describes the ICA method. Referencing methods are presented in Nam et al. (2018). Blankertz (2018) presents the full pipeline from continuous EEG data to classified output, including bandpass filtering and epoching.

4.2 Feature extraction

The second step to decode brain activity is feature extraction. The features are extracted from the cleaned epochs from the preprocessing step (see Section 4.1). The features are then used for classification, the third and final step to decode the brain activity (see Section 4.3). This section will look at some feature extraction methods for EEG data.

The purpose of feature extraction is to facilitate classification. The machine learning algorithm finds patterns in the data and uses these patterns to classify the brain activity. The challenge is that data from brain activity, such as EEG data, is complex and the patterns are difficult to extract. Thus, feature extraction is needed to highlight important data characteristics and reduce the data's dimension.

The choice of feature extraction depends on the BCI paradigm and machine learning method. Generally, for the P300 paradigm, the average of the EEG data from multiple trials is the prominent feature. For SSVEP, the frequency information in the EEG data is the most important feature, and for motor imagery, the difference in spatial activity of the EEG data is important. However, each BCI setup is different, and you must carefully consider what type of feature extraction is best for your case.

In this section, I present some common feature extraction methods, such as principal component analysis, common spatial patterns, and covariance matrices. It is not a complete list of all possible choices, but it should be enough to guide you through most papers. One should also remember that the feature extraction method should work well with the machine learning method.

Averaging the EEG data

The first feature extraction method we will look at is the simple averaging of EEG signals. From Section 3.2, we know that in the P300 paradigm, it is possible to see if the user attended the given stimuli or not by looking at the response in the EEG data around 300 ms after stimuli onset. The EEG data is averaged over multiple trials for each class to get signals with less noise, which, in the end, results in a slow classification rate since it takes time to run multiple trials. For the P300 paradigm, averaging the EEG signals is often enough, but that is not the case for the other paradigms.

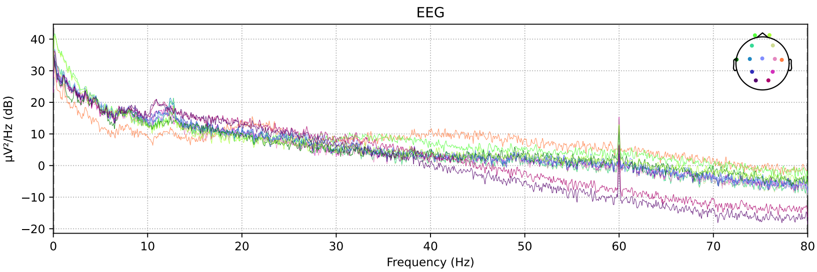


Figure 4.3 Power Spectral Density from Motor imagery EEG data. The x-axis shows the frequencies, and the y-axis shows the power. One power spectra per EEG channel. 60 Hz power grid artifact is visible in all channels.

Power Spectral Density (PSD)

The second feature extraction method is the Power Spectral Density (PSD), which shows the frequency components of the EEG signal. The frequency components (power of the frequency) tell "how much" of each frequency there is in the data. Usually, the PSD is calculated for each channel (electrode) individually and plotted in a diagram with frequency on the x-axis and power on the y-axis (see Figure 4.3). Some variant of the Fourier transform is usually used to transform the EEG signal to PSD.

A high peak can be seen in the PSD at 50 Hz (or 60 Hz in, e.g., the USA) before the power grid artifacts are removed. Sometimes, there are differences in the PSD between electrodes, such as peak power (how high the peaks are) or peak frequency (where along the x-axis the peak is located). In those cases, the PSD could be a good feature to use for further classification.

Time-frequency

Instead of looking at the power spectra for the full signal, as in PSD, one can look at the power spectra over time in time-frequency plots. Looking at the power spectra over time gives more information about when the brain activity was happening. The time-frequency plots have time on the x-axis, frequencies on the y-axis, and the color shows the power. A time-frequency plot is basically multiple PSD plots for small time-chunks stacked horizontally (see Figure 4.4).

There are different methods to generate time-frequency data, and each method has parameters to tune how the time-frequency analysis is done. The short-term Fourier Transform (STFT) was one of the early suggestions. STFT splits the signal into successive windows and performs the Fourier Transform on those to generate power spectra over time. Another method is the wavelet transform (WT), which is a type of template matching. Wavelets catching different frequencies are generated from the mother wavelet and convoluted with the brain signal to generate the time-frequency data. The mother wavelet can be designed in multiple ways to catch

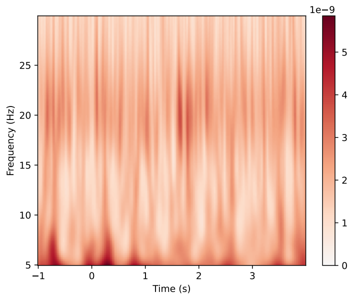


Figure 4.4 Time-Frequency plot from Motor imagery EEG data from Cz electrode. The x-axis shows the time, the y-axis the frequency, and the color the power.

different features of brain activity.

By comparing different electrodes and patterns in the time-frequency domain, one can find features suitable for classifying brain activity.

Covariance matrices

Moving on from frequency-based features, the following feature extraction method we will investigate represents the EEG data as covariance matrices. The covariance matrix of data captures the data's correlation and "spread". For EEG data, a covariance matrix tells how correlated the EEG signals in different electrodes are. If there is a lot of activity in electrode i, the covariance matrix tells if there will also be a lot of similar activity in electrode j.

The EEG data for an epoch is $x \in \mathbb{R}^{m \times n}$, where m is the number of channels (electrodes) and n is the number of time samples. Thus, $x_i \in \mathbb{R}^m$ is the EEG data for all electrodes in a single time sample. The covariance matrix $\mathbf{S} \in \mathbb{R}^{m \times m}$ for an epoch is calculated as

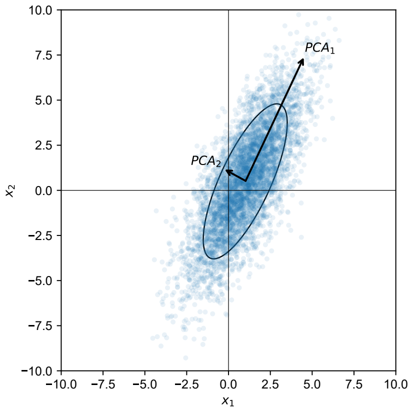
$$\mathbf{S} = \frac{1}{n} \sum_{i=1}^{n} (x_i - \overline{x})(x_i - \overline{x})^\top, \tag{4.2}$$

where $\overline{x}$ is the mean calculated as

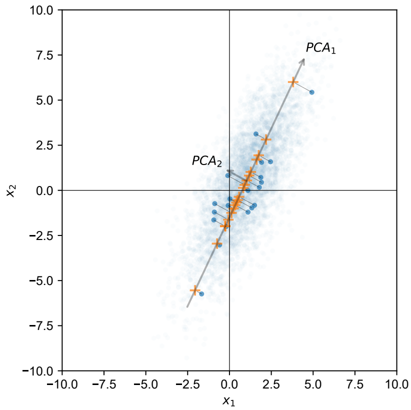
$$\overline{x} = \frac{1}{n} \sum_{i=1}^{n} x_i. \tag{4.3}$$

Covariance matrices are symmetric, positive semidefinite matrices, which give them a lot of useful properties.

The covariance matrix is an intermediate step in many methods, such as PCA and CSP. But the epochs' covariance matrices can in themselves be used as features for classification, especially for Riemannian-based classifiers (see Section 4.3 for more details on Riemannian-based classifiers).



(a) The image shows the original two-dimensional data with an ellipsoid overlay showing the distribution. Principal components are shown as arrows.



(b) The image shows the data projection onto the first principal component. Blue (o) markers show the original data and orange (+) the projected data.

Figure 4.5 Illustration of PCA on two-dimensional data.

Principal component analysis (PCA)

The next feature extraction method is the principal component analysis (PCA), which is an old dimension reduction technique with two applications in the BCI area: dimension reduction and artifact removal.

The idea behind PCA is that high dimensional data have some directions with much information and others with little information. The directions without much information can be removed altogether, and thus, the dimension of the data has been reduced.

Looking at illustrative data in two dimensions (see Figure 4.5a), we can see that the data is most distinguishable in the direction pointing to the top right corner (PCA_1). In simple words, if you were to describe a data point you would say how far away in the direction of PCA_1 it was placed. The direction PCA_1 is called the first principal component of the data and corresponds to the direction in the data with the highest variance. In PCA, it is assumed that PCA_1 is the direction with the most important features of the data, PCA_2 is the direction with the second most important features, and so on until PCA_n where n is the dimension of the original data. Thus, we can reduce the dimension of the data by removing principal components with little information. Little information means, in this case, small variance. In the illustrative two-dimensional example, there are only two principal components, and the data's dimension can be reduced by removing PCA_2 (projecting the data onto PCA_1) (see Figure 4.5b).

The first step in PCA is to find the directions (principal components) of the data with the highest variance, and the second step is to project all data onto these principal components. The covariance matrix $\mathbf{S}$ of the data $x \in \mathbb{R}^d$ describes the variance in the data. The principal components are found as the k eigenvectors ($k < d$) with the highest eigenvalues from the covariance matrix. The data is linearly transformed into the principal components. Thus, the dimension of the data is reduced.

A limitation of PCA is that it assumes that the interesting information is found in the directions with high variance, which is not guaranteed. Even so, PCA has proven to be an effective dimension-reduction method for BCIs. PCA has, in BCIs, also been used for artifact removal. Artifact removal is possible when the artifact is uncorrelated with the brain signals because then the artifacts end up in a principal component of their own that can be removed altogether.

Common Spatial Patterns (CSP)

The final feature extraction method I will describe in this thesis is the Common Spatial Patterns (CSP). The idea behind CSP is to transform the EEG data from the original sensor space $x(t)$ to the "CSP space" $x_{csp}(t)$ via the transformation matrix W as

$$x_{csp}(t) = W^\top x(t). \tag{4.4}$$

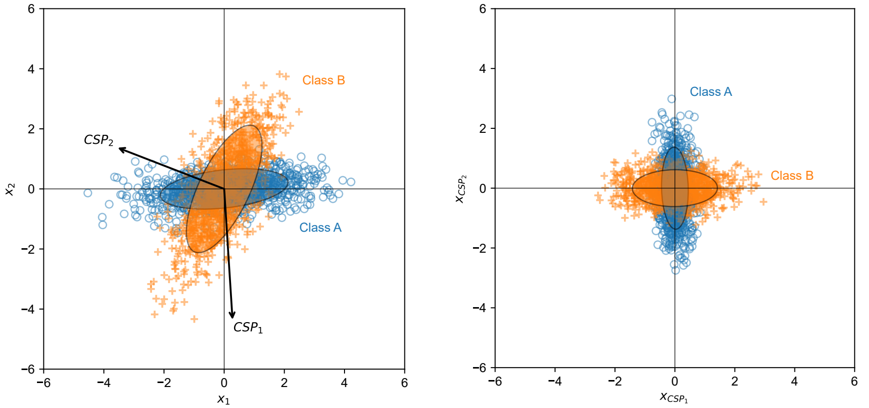
The "CSP space" $x_{csp}(t)$ is designed to have features that make the classes easy to separate and classify. The CSP algorithm finds the transformation matrix W by looking at the classes' mean covariance matrices and finding directions where the classes are the most separable. In CSP, it is assumed that a direction where one class has much variance and the other class has little variance is a direction where the classes can be separated from each other. CSP was originally designed for data with two classes but has been extended for more classes.

Looking at illustrative data in two dimensions, we see that the new features (x_1 and x_2 in Figure 4.6b) make classification easier (see Figure 4.6). Four to six CSP features are often chosen to describe the data in a BCI setting. Thus, CSP also works as a dimension-reduction method.

The transformation matrix W mixes the EEG data from different channels to generate the new CSP data $x_{csp}(t)$. Because of this, CSP is a spatial filtering method. Spatial filters can be viewed as topographic maps, which are heatmaps over the electrodes showing where the brain activity in the CSP features originate from. Figure 4.7 shows the topo maps for the first four CSP features from some motor imagery data. We note that CSP_0 and CSP_2 capture activity in the left- respectively right half of the brain and could be used to classify left- vs. right-hand movement.

Section summary and References

Summary. In this section, we have taken a closer look at the feature extraction step, which is the second step to decode EEG data. Some commonly used feature extraction methods are covariance matrices, PCA, and CSP. In the next section, we



(a) The image shows the original two-dimensional data. CSP basis plotted as arrows.

(b) The image shows the data after the CSP transform. One can see that the two classes blue (o) and orange (+) are aligned in such a way that classification is simplified.

Figure 4.6 Illustration of CSP transformation on two-dimensional data belonging to two classes: Blue (o) and, orange (+). The ellipsoid overlays show the data distributions.

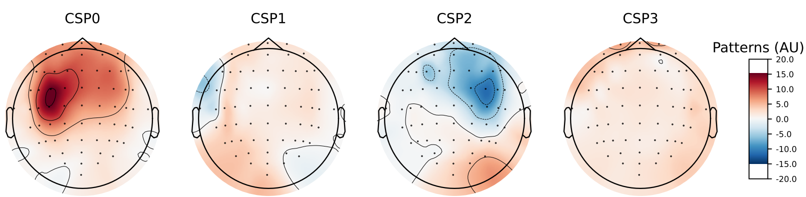


Figure 4.7 Spatial patterns used for CSP transformation of EEG data from a Motor Imagery experiment. The topo maps show the spatial filters CSP uses to extract the CSP features. We see that CSP_0 and CSP_2 catch left, respectively, right brain half activity.

will look at the third and final step to decode brain signals, namely classification with machine learning.

References. The paper by Nicolas-Alonso and Gomez-Gil (2012) thoroughly explains different feature extraction methods. Blankertz (2018) shows how averaging can be used as a feature extraction method. Faisal et al. (2020b) explains the mathematical details of covariance matrices. Faisal et al. (2020a) gives an in-depth explanation of PCA. Blankertz et al. (2008) provides a thorough explanation of CSP.

4.3 Machine Learning

The final step to decode brain activity is to classify the data with a machine learning algorithm. In this section, I present some common machine learning algorithms for BCIs. It is by no means a complete list of all existing machine learning algorithms, nor all possible machine learning algorithms for BCIs, but I present some of the most commonly used algorithms for BCIs.

A machine learning algorithm can be used to solve many types of problems, for example, classification and regression problems. In a *classification* problem, the aim is to predict the class of new data into predefined classes. In a *regression* problem, the aim is to predict a numerical value, e.g., a person's age. BCIs are usually a classification problem, e.g., what hand was the user imagining to move. BCIs with regression problems also exist, e.g., what coordinates on the computer screen is the user trying to move the cursor to, or the user's alertness level. Since the classification problem is the most common for BCIs, I will hereafter assume that the machine learning algorithm is used for classification.

Data with known labels is used to train the machine learning algorithm. The labels state which class the data belongs to (or the correct numerical value in the regression problem). One issue is that a machine learning algorithm can be overfitted, which means that it can perfectly predict the label of the training data but cannot generalize to new data. The goal is often for a machine learning model to generalize to unseen data. Usually, when training a machine learning algorithm, one uses some training data for validation. The validation data is only used to see how well the model generalizes and to select between different models or hyperparameters (parameters for the machine learning model) and not for training the machine learning algorithm. Test data is used for the final testing of a model and not at all used during the training.

When training a machine learning algorithm with training data, the algorithm compares the predicted label with the true label for the data. Then, the algorithm tweaks its parameters to improve the prediction. It uses a cost function (also called loss function) to compare the predicted label and the true label. When evaluating the classification performance, we often use the *accuracy*, which states the ratio of correct classifications.

This section presents some commonly used machine learning algorithms in BCIs. I also introduce Riemannian geometry-based classifiers, which are state-of-the-art for many BCI applications, and present transfer learning for BCIs. The term classifier means a machine learning algorithm used for classification.

k-nearest neighbors classifier (k-NNC) and Minimum distance to mean (MDM)

We will start by looking at two simple classification methods: k-nearest neighbors classifier (k-NNC) and minimum distance to mean (MDM).

The nearest neighbor classification method classifies a new data point to the same class as the k closest neighbors. It is as simple as looking at the class for the k closest neighbors, and if they belong to different classes, the class to which most neighbors belong is the class for the new data point.

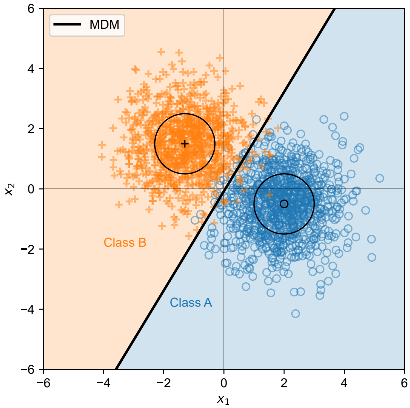
The MDM calculates the mean for all classes and classifies a new data point to the same class as the closest mean. In a two-dimensional example with two classes, one can view this as drawing a line, a so-called hyperplane, between the means and classifying the data depending on what side of the line it is (see Figure 4.8a).

Both these two classifiers are simple but have proven powerful for BCIs when used with Riemannian geometry, especially MDM. See Section 4.4 for more details on Riemannian Geometry.

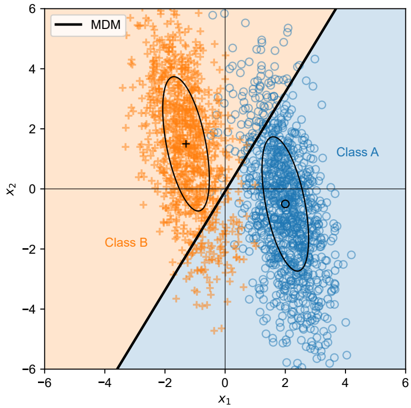
Linear Discriminant Analysis (LDA)

Depending on the data distribution, the simple approach of MDM might not be enough. This is illustrated in Figure 4.8b where we can see that the MDM approach misclassifies many samples from both classes due to the data distribution. Linear discriminant analysis (LDA) uses the covariance matrix of the data to take the data distribution into account when finding a separating hyperplane between the classes (see Figure 4.8c).

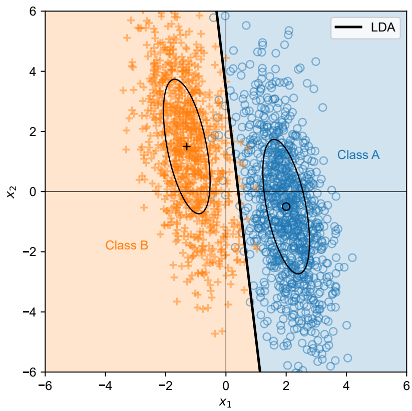
A drawback is that LDA assumes that all classes have the same distribution. Some LDA variants, e.g., Quadratic Discriminant Analysis, can handle data with different distributions. Another drawback is that LDA requires a lot of data for correctly estimating the covariance matrices (estimating the distributions). Suppose the estimation of the covariance matrices (the distributions of the data) is bad. In that case, assuming a distribution based on prior knowledge is often better than using the estimation. A covariance shrinkage method estimates covariance matrices by mixing covariance matrices based on prior knowledge and covariance matrices from the current data. Covariance shrinkage has proven very powerful in BCI applications where the amount of data is limited, which, without covariance shrinkage, results in bad covariance estimations.



(a) MDM classification of data with a diagonal unitary covariance matrix.



(b) MDM classification of data with correlation between the features (non-unitary covariance matrix).



(c) LDA classification of data with correlation between the features (non-unitary covariance matrix).

Figure 4.8 Classification of two-dimensional data belonging to the two classes blue (o) and orange (+). The drawn line represents the classification boundary, and the shaded background represents the predicted class. The ellipses show the distribution of the data.

Logistic regression

The next machine learning algorithm is the Logistic regression, which, despite the name, is used for classification. It outputs the probability p that a data point belongs to a specific class as

$$p = \sigma(x^\top \theta), \tag{4.5}$$

where x is the data, θ is the parameters for the algorithm, and $\sigma(\cdot)$ is the logistic (sigmoid function), which outputs a number between 0 and 1. If the probability p for a data point is above 0.5 (50%), the predicted class is 1 otherwise the predicted class is 0. The parameters θ are tweaked during training so that the returned probability for the training samples moves towards 0 and 1 (depending on the label). In other words, the parameters are tweaked so that the classifier gets more sure about the prediction of training data.

Equation (4.5) is for the two-class case, but logistic regression also works for multiple classes and is then called Softmax regression. The principle for softmax regression is the same as logistic regression but adapted for more classes.

Even though logistic regression is more advanced than the simple MDM, it is still a relatively simple classification algorithm.

Support Vector Machine (SVM)

The next machine learning algorithm, Support Vector Machine (SVM), is slightly more advanced than the previous ones.

The intuition behind SVMs is to find a separating hyperplane with as much margin to the data as possible. In other words, SVM tries to fit an as wide as possible road between the data (see Figure 4.9). Due to this margin, the SVM often performs well and is less prone to overfitting. The data in Figure 4.9 is separable with a hyperplane with margins. On the other hand, the SVM method can accept so-called margin violations if the data were to be a bit more mixed in the middle. A parameter in SVM can be used to tune its sensitivity to margin violations.

SVM is a popular machine learning algorithm in general but has also been used in BCIs. SVM is, by default, a linear classification method but can tackle nonlinear problems with the so-called "kernel trick". The kernel trick transforms the data in a nonlinear way but can then, under the hood, do linear classification. There are many different kernels, e.g., the polynomial kernel, which generates polynomial features. The kernel generally used in BCIs is the Gaussian radial basis function (RBF), which is a 'similarity' kernel that generates features indicating how similar the data is.

Artificial Neural Networks (ANN)

The next machine learning algorithm, Artificial Neural Networks (ANNs), is even more advanced than Support Vector Machines. The field of ANNs which includes Deep Neural Networks (DNNs) and Convolutional Neural Networks (CNNs) is huge. I recommend part two in the book Géron (2019) for the interested reader.

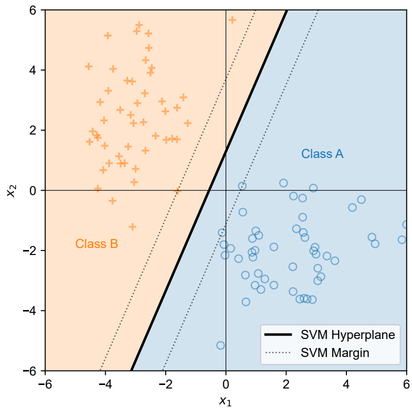


Figure 4.9 Classification of two-dimensional data belonging to the two classes blue (o) and orange (+) using SVM. Full line (-) is the decision boundary and dotted lines (:) are the margins in SVM. The shaded background shows the predicted class.

I want to emphasize that I here give a brief overview of the basic concepts but there is a vast amount more to explore.

The smallest unit in an ANN is the *perceptron*, also called neurons, units, or nodes. It takes inputs that it weighs together and uses an *activation function* to calculate its output. A neural network is built of perceptrons in multiple layers. A neural network always has an input layer, an output layer, and often one or more hidden layers in between (see Figure 4.10). The number of hidden layers and the structure is what differentiates DNNs and CNNs from simple ANNs. The magic behind ANNs is the backpropagation algorithm, which calculates how to improve all weights in all perceptrons.

There are many different activation functions, and their task is to do a nonlinear transformation of the perceptron's inputs. Some common examples are ReLU, SELU, ELU, sigmoid, and softmax. When using a neural network for classification, you commonly put a logistic regression or softmax function on the output layer (logistic for multilabel binary classification, softmax for multiclass classification). The functions transform the output to the range of 0 to 1, which can be interpreted as class probabilities.

Convolutional Neural Networks (CNNs) are a popular variant of neural networks. CNNs were primarily used for images but have also been used for EEG data. The idea behind CNNs is to have lower layers with units that look for a specific pattern in a small part of the data, e.g., a corner or a straight line in image recognition. At the higher levels of the CNN, these patterns are combined into bigger features, e.g., an eye or a house. CNNs generally require fewer parameters to recognize an image than a fully connected neural network. However, they still have a lot of parameters and require a lot of data to be trained.

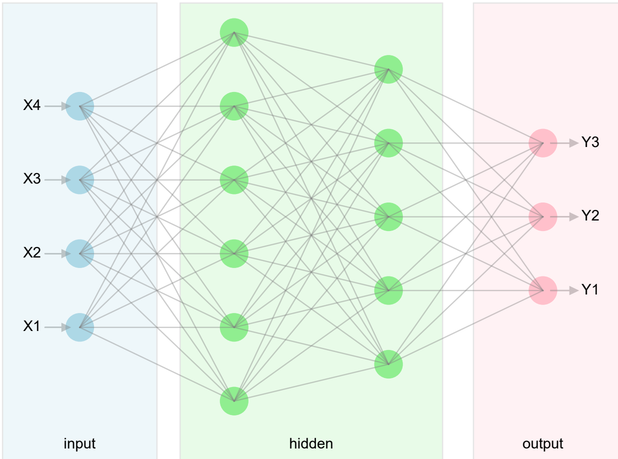


Figure 4.10 Schematics of a neural network with four nodes in the input layer, two hidden layers with six respectively five nodes, and one output layer with three nodes.

How to choose a classifier?

Now that we have seen many different classifiers, the next obvious question is "How to choose a classifier?".

Unfortunately, there is no golden advice to give about choosing classifiers. It all boils down to what BCI paradigm was used, what preprocessing was done, what features were extracted, what performance requirements exist, the data quality, how much data is available, and a hundred other factors. My best recommendation is to look at the two papers by Lotte [Lotte et al., 2007] and [Lotte et al., 2018a], which compare different machine learning algorithms that have been used on BCI data.

Section summary and References

Summary. In this section, we have learned about the third and final step to decode brain activity, namely the classification step. We have looked at different machine learning algorithms for classifying BCI data, such as Minimum Distance to Mean, Logistic Regression, and Support Vector Machines. In the next section, I introduce Riemannian geometry-based classifiers, which are state-of-the-art for BCIs.

References. The two papers by Lotte [Lotte et al., 2007] and [Lotte et al., 2018a] thoroughly review the classification methods and transfer learning for EEG-based BCIs. Nearest neighbour is explained in Nicolas-Alonso and Gomez-Gil (2012) and LDA is explained in detail in Blankertz (2018). The excellent book by Géron (2019) contains both math explanations and hands-on examples of different machine learning methods, e.g., logistic regression, SVM, and neural networks. Most machine learning algorithms are implemented, e.g., sci-kit learn for Python [scikit-learn developers, 2023], accompanied by detailed documentation of the methods.

4.4 Riemannian Geometry

You are probably familiar with the so-called flat Euclidian space taught in most basic math courses. In a flat space, properties such as angle and length behave as "expected" and are defined by the inner product of the space. In a so-called curved space, on the other hand, these properties behave differently and are defined by the metric tensor. Euclidian geometry describes flat spaces, and Riemannian geometry can describe most curved spaces.

An example of a flat space is the surface of a table and an example of a curved space is the surface of a sphere, the Earth for example. Figure 4.11 compares flat (middle) and curved (left and right) spaces and illustrates the following properties of the spaces:

- **Lines** – In all spaces, a line is the segment between two points that follows the structure of the space.
- **Circumference of circle** – Given a circle with the diameter $d = 1$ (along the surface). In a flat space, the circle's circumference is $C = d\pi = \pi$, while in a curved space, the circumference is generally $C \neq \pi$.
- **Parallel lines** – Parallel lines never intersect in a flat space. A line perpendicular to the parallel lines crosses them at a 90° angle. In a curved space, on the other hand, two lines that are seemingly parallel at one point will cross or diverge at another point (depending on the curvature of the space).
- **Sum of angles of triangles** – The sum of angles of triangles is 180° in a flat space. however, in a curved space, the sum could be more or less than 180° (depending on the curvature of the space).

Since objects in a curved space don't behave as expected a metric tensor is needed to specify how these measures are defined. A so-called Riemannian manifold, or Riemannian space, is a curved space that can, at every point of the space, be locally approximated with a Euclidian tangent space. The local approximation is accompanied by a metric tensor specifying how, e.g., lengths and angles are defined. Another important concept of a Riemannian manifold is parallel transport which defines how a vector in the tangent space at one point of the manifold is transported to the tangent space of another point on the manifold. The Levi-Civita connection defines how to do torsion-free parallel transport (the result of parallel transporting a vector $\vec{v}$ along $\vec{u}$ is the same as transporting $\vec{u}$ along $\vec{v}$) and keeping the inner product of two transported vectors the same. There are a lot of mathematical details related to Riemannian geometry that I skip here, my goal is to give a basic intuition for the

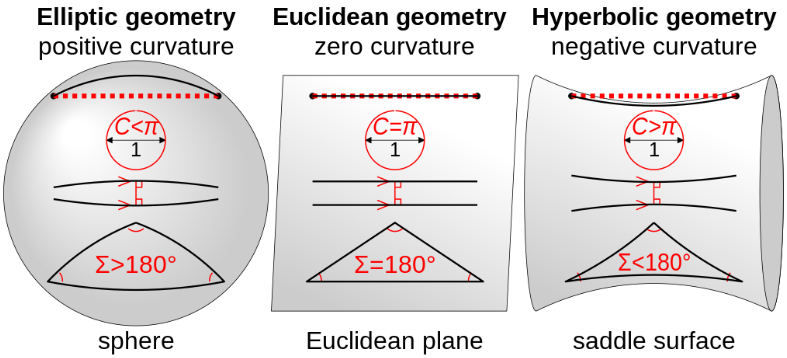


Figure 4.11 Comparision of different geometries and how these affect the concept of lines, the circumference, parallel lines, and the sum of angles of triangles.

concepts. I highly recommend the video series by Eigenchris on Tensor Calculus for the interested reader (see the references for this section).

There are subcategories of the Riemannian manifold, and each comes with its own way of measuring distances, angles, and so on. The covariance matrices from BCI data with a given metric tensor belong to a particular Riemannian manifold of symmetric positive definite matrices.

Data on a Riemannian manifold can either be classified on the Riemannian manifold or the data can be projected to a Euclidian tangent space of the manifold before classification. On the Riemannian manifold, e.g., minimum distance to mean can be used for classification, with the distance between data points defined by the metric tensor. In the tangent space, any 'normal' Euclidian classification method can be used, such as logistic regression, SVM, or LDA. Riemannian geometry-based classifiers are state-of-the-art for many BCI applications.

Section summary and References

Summary. In this section, we have introduced Riemannian geometry-based classifiers, which are state-of-the-art for BCIs. In the next section, we will talk about transfer learning.

References. The excellent videos by "Eigenchris" [eigenchris, 2020; eigenchris, 2017] describe tensor calculus in detail. Barachant et al. (2012) introduced Riemannian Geometry to BCIs.

4.5 Transfer learning

Transfer learning is not a machine learning algorithm but a tool for machine learning when we don't have access to enough training data.

When training a machine learning algorithm, it is assumed that the training and test data come from the same distributions. In the BCI case, training data is what we collect in the calibration phase to train the machine learning algorithm, and test data is what we use when the BCI system is in operation. When we say that the training and test data come from the same distribution, it means that the BCI system is calibrated right before it is used, i.e., the training data and test data come from the same session. Since the calibration phase takes a relatively long time, it would be preferable to use data from a previous session for calibration so that the user can immediately use the system instead of starting with calibrating the system. This is where transfer learning comes into play.

The idea with transfer learning (framed in the BCI application) is to use data from the user's previous sessions or other persons' previous sessions to train the BCI system for a new session. In the language of transfer learning, the previous data is the *source domain*, and the new data is the *target domain*. There is a myriad of different approaches to performing transfer learning. Here I present some common approaches for BCIs:

- **Select data where source and target domain coincide** – A simple approach for transfer learning is to use only data from the source domain that coincides with the data in the target domain. Assuming that there is a pool of source data to use, e.g., data from different users, it would be possible that some, but not all, of this source data coincide with the target data. If only data that coincide is used, no data transformation is needed. The challenge with this approach is identifying the data in the source domain that coincides with the data in the target domain.

- **Find common features in all domains and then train a classifier on these features** – Another transfer learning approach is identifying features that coincide in all domains. It means that no data transformation is needed, but it might also mean that important features are ignored simply because they don't coincide. In the best of worlds, the common features represent the true brain activity and would be the best features to use.

- **Transform data in source domain to target domain** – A third approach for transfer learning is to transform the data in the source domain to fit the data in the target domain. One way to do this is to move the covariance matrices representing the data on the Riemannian manifold. The source domain data is moved so that the source domain data's mean is moved to the target domain data's mean.

- **Adapt models from the source domain to the target domain** – A final approach to transfer learning is using a machine learning algorithm in the target domain with statistically likely parameters based on source domain training. Put the other way around, a specific machine learning algorithm (e.g., logistic regression) is trained for each dataset in the source domain. Then, the distribution for each parameter in the machine learning algorithm can be found, and the statistically most likely value for each parameter (e.g., the mean, mode, or median) can be used for the model used on data in the target domain. Thus, a classifier based on the parameters in the source domain can be used to classify data in the target domain.

Section summary and References

Summary. In this section, we have learned about Transfer learning as a tool for classification. In the next section, I list some useful Python packages when working with BCIs.

References. Transfer learning for BCIs is well explained in Jayaram et al. (2018) and Lotte (2015).

4.6 Python tools for working with EEG-based BCIs

This section is a list with links to good tools to use when working with EEG data. They are Python packages since I have chosen to use Python so far. Many packages are available, but those I present here are the ones I have used. I omit basic packages such as Numpy, Pandas, Matplotlib, etc.

- **MNE**[1] – MNE is a Python package for analyzing and visualizing EEG data. It includes preprocessing tools and some feature extraction tools.
- **MOABB**[2] – Mother of all BCI Benchmarks is a Python package that includes a lot of public available datasets and some tools for comparing the performance of a classifier across different datasets. It is a package under development, and new features are added frequently. I have mainly used it to access public available datasets.
- **Scikit-learn**[3] – Scikit-learn is one of the most significant Python packages for machine learning. It includes algorithms related to machine learning.
- **PyRiemann**[4] – PyRiemann is a package for Riemannian geometry.
- **Timeflux**[5] – Timeflux is a Python package for real-time processing of biosignals. Typically useful to build a BCI.

[1] https://mne.tools/stable/index.html
[2] http://moabb.neurotechx.com/docs/index.html
[3] https://scikit-learn.org/stable/
[4] https://pyriemann.readthedocs.io/en/latest/
[5] https://timeflux.io/

- **PsychoPy**[6] – PsychoPy is a Python package for creating experiments. I have used it to create stimuli programs.
- **BCI-HIL**[7] – BCI-HIL is the framework for designing BCIs presented in Paper II, see Chapter 7 and Paper II for details.

Section summary and References

Summary. This short section lists Python packages I used for EEG data. The next session is an outlook into the future of algorithms for BCIs.

References. The links to each package are found in the footnotes at the bottom of the page.

4.7 Comparision of the development of BCIs and computer vision

We have now seen some machine learning algorithms that are used today in BCIs. To get a glimpse of what methods could be available in the future, I will compare the development of classifying BCI data to the development of classifying images, which is a fundamental part of computer vision.

In simple terms, BCIs' goal is to classify brain activity, and computer vision's goal is to classify an image's content. Classification of BCI data requires extensive preprocessing and feature extraction. Few off-the-shelf pipelines can be used, so much expertise is required to classify the BCI data. For image classification, on the other hand, there exist several off-the-shelf machine learning algorithms that do it for you. You input your image and get the classification as output. The difference in effort between BCI classification and image classification is striking. The interesting thing to note here is that image classification in its early days also required a lot of expertise and manual feature extraction.

To classify an image, one needs to identify the features of the image, e.g., where the lines in the image are, how these lines are connected, what structures these connected lines create together, and so on, until you eventually can say that this image shows a house. In the early days of computer vision, these features were manually extracted. When I say manually extracted, I mean that algorithms are applied to the image, e.g., to identify lines, in a similar way as we apply algorithms to preprocess and do feature extraction on EEG data today. In computer vision today, all these features are extracted as part of the machine learning algorithm. What happened between the early days of computer vision and today was that the Convolutional Neural Network (CNN) was developed. It is a machine learning algorithm that aims to find smaller features of an image, which are then combined into higher-level

[6] https://www.psychopy.org/
[7] https://www.bci.lu.se/bci-hil

features that eventually can be used for classification – all done by the CNN. Another thing that happened to the development of computer vision was that massive datasets of images were produced, which made it possible to train advanced machine learning algorithms such as CNNs.

Looking at the tremendous improvement of computer vision, it is not too radical to guess that something similar will happen to BCI in the future. More and more BCI data will become available, which allows the development of machine learning algorithms tailored for BCIs. Once we have these powerful machine learning algorithms for BCIs, we can reduce or even remove the calibration of BCIs altogether. Until then, we will have to keep working with handcrafted features, transfer learning, and other clever tools to reduce the calibration time for BCIs.

Section summary and References

Summary. In this section, we compared BCI classification's development to computer vision's development. The following section is a summary of the whole chapter.

References. The paper, [Bhatt et al., 2021], discusses the history of computer vision and the development of CNNs.

Chapter summary

In this chapter, we have learned how to decode brain signals.

Section 4.1 explains the first step to preprocess the EEG data, which includes artifact removal, bandpass filtering, and epoching.

Section 4.2 explores the second step, which is to extract features from the epochs. Feature extraction highlights important aspects of the data and reduces the data dimension. Common methods for feature extraction are PCA, CSP, and covariance matrices.

Section 4.3 describes the final step to decode brain signals, which is to classify the data with a machine learning algorithm. For BCIs, the state-of-the-art methods are Riemannian geometry-based classifiers (Section 4.4), but other common classifiers are LDA, SVM, and neural networks.

Section 4.4 introduces Riemannian geometry which many of the state-of-the-art machine learning methods are based on.

Section 4.5 describes transfer learning and highlights some commonly used approaches for BCIs.

Section 4.6, lists some useful Python tools when working with BCIs.

Finally, Section 4.7 gives an outlook for future machine learning algorithms for BCIs by comparing the development of BCIs to that of computer vision.

In the next chapter, I will highlight some ethical concerns related to BCIs.

5

Ethics

In the previous chapters, we have learned the details of how Brain-Computer Interfaces work. The research on BCIs enables many interesting future uses of BCIs. However, there are a lot of ethical concerns to consider before we are there.

This chapter presents some commonly raised ethical concerns related to BCIs. I do not offer any solutions since these concerns are still being investigated by the research community. I focus on the technical aspects but recommend that interested readers consult the provided references for other ethical aspects.

- **Privacy and Data Security** – The first ethical concern that usually surfaces is privacy and data security. How can a user be guaranteed that their data does not end up in the hands of a sinister government, company, or person? And how can the user be guaranteed that only information used for the BCI, and not other personal information, is extracted? Another concern is what could happen if the BCI systems are hacked while in use.

- **Data bias** – A second ethical concern regards data bias. As always, when working with data and models trained on data, there will be a bias toward the training data. Data similar to the training data will perform better with the model than dissimilar data. As an example of data bias, we can take the case of voice recognition. If a model for voice recognition is only trained on male voices, it will probably have a hard time recognizing female voices. Data bias in the context of BCIs could be related to features of the users, such as age, gender, IQ, size/anatomy of the head, or other unknown factors.

- **Responsibility** – Another ethical concern is related to the responsibility of the BCI systems. When we use BCI systems in our everyday lives, the question arises: "Who is responsible for the outcome of the system?". The user or the one designing the system? This is a comparable ethical dilemma to the one for self-driving cars. Is the driver or the one who created the vehicle responsible for the accidents?

- **Superhumans** – A lot of BCI research is focused on improving the lives of disabled persons, such as communication for locked-in patients or prosthetic limbs for persons who are missing a limb. With this focus, there is the underlying assumption that the life of a disabled person is less than a “normal” life and thus needs to be fixed. As you see, this whole formulation is problematic. There is also the question of whether using BCIs (e.g., for prosthetics) can create superhumans with abilities that surpass the normal.

- **Availablility** – As with all new technology, the question arises of who can afford and use the technology. New technology tends to cost a lot of money. Thus, only the rich can afford it.

This list of ethical considerations does not cover all ethical concerns related to BCIs but showcases some of the commonly raised concerns for BCIs.

Section summary and References

Summary. This short chapter presented some ethical concerns related to the development of BCIs. The next chapter reviews the calibration challenge and motivates the problems I study in my research.

References. The paper Burwell et al. (2017) thoroughly examines the ethical aspects of BCIs and how these are discussed in the literature. The paper Saha et al. (2021) presents some challenges and opportunities in BCIs and raises ethical concerns relating to these topics.

6

Problem formulation

The core of my research is to improve the calibration of Brain-Computer Interfaces (BCIs). Until now, in the thesis, you have gotten a lot of background to BCIs and an introduction to state-of-the-art methods. Chapter 2 introduced BCIs, Chapter 3 explained how the brain works and how brain signals can be measured (focusing on EEG), Chapter 4 described state-of-the-art methods to preprocess, extract features, and classify the EEG data, and finally Chapter 5 highlighted some ethical concerns related to BCIs.

Throughout the thesis, I have emphasized the challenges with the calibration of BCIs and how these limit the applications of BCIs. In this chapter, I will review the calibration challenges and look at current methods to improve the calibration of BCIs.

6.1 Reviewing the problem with calibration

As you might have noticed, the challenges and limitations of calibration appear repeatedly when talking about BCIs. I want to clarify that there are other challenges and limitations with BCIs than the calibration. I choose to highlight those related to the calibration of BCIs since this is where my research focus lies.

From the previous chapters, we know that the brain activity and EEG data are slightly different for every session due to different mental states, EEG electrode placements, and other ongoing brain activity. Thus, the BCI must be (re-)calibrated every time before it can be used. We have seen that calibration includes collecting labeled data from the user and training a machine learning algorithm. Data is collected while a stimuli program instructs the user on what to think about (e.g., imagine moving the right hand). Calibration of BCIs is often tiresome for the user due to the long time it takes to collect data, but we know that calibration is necessary for the BCI system to work - the machine learning algorithm in the BCI needs to learn the patterns in the user's brain activity. Generally, the classification gets better the more data is available. So, there is a tradeoff between collecting a lot of data and tiring the user. Collecting a sufficient amount of data to train a machine learning algorithm can take up to 20 minutes. We have also seen in previous chapters that accepting long calibration times and performance requirements depends on the application. The long and tedious calibration of BCIs is one of the reasons why we don't see any BCI applications in our everyday lives. It simply takes too long to calibrate the system to be worth using it. There are, of course, other reasons why we don't use BCIs in our everyday lives.

To summarize, the purpose of calibrating a BCI system is to adjust the BCI for a new session, and it is often a long and tedious process for the user.

6.2 Tackling calibration challenges

The takeaway from the calibration challenges is that little data from a session is available for training a classifier. Preferably, even less data should be collected during the BCI calibration to ease the user experience. Thus, the problem is to train a classifier with little data even though a lot of data is needed. There is a lot of research on how to solve this problem. Below is a list of some approaches:

- **Regularization** – One approach is to use regularization on the algorithms. Ordinary regularization methods of machine learning algorithms (such as lasso regularization, ridge regularization, or elastic net regularization) are used to avoid overfitting the data. Another type of regularization is to impose parameters for the algorithm that are known to perform well. Covariance shrinkage is an example of this that we discussed in the part about LDA (see Section 4.3).
- **Adaptive learning, Unsupervised learning, or Semisupervised learning** – A second approach would be to update the machine learning algorithm while the BCI is in use. A challenge is that the data we get while the BCI is in use is unlabeled. Even so, the data could still be used to update the machine learning algorithm. Depending on the specifics of how this is done, we call it adaptive learning, unsupervised learning, or semisupervised learning.
- **Transfer learning** – Another approach is to use previously collected data and do transfer learning as we discussed in Section 4.5. Then, more data would be available for training the machine learning algorithm without the need to collect more data during the calibration of the BCI. With transfer learning, it could even be possible to create a BCI system that only uses old data and does not need to be calibrated at all. However, more tools for transfer learning are available if some calibration data is collected, compared to when no data is collected at all.
- **Features** – A final solution would be to find features that better highlight the brain activity. It is impossible to predict what these features would be, but one possibility is that the signal reconstruction of brain activity is improved so that the features are the true brain activity. Another possibility is

that machine learning algorithms tailormade for BCIs are developed. These BCI machine learning algorithms could find the interesting features themselves and maybe even do the preprocessing of the EEG data. This futuristic view of BCI-machine learning algorithms is comparable to the development of computer vision, which I expanded on in Section 4.7.

Even though there are many approaches to overcoming the calibration challenge of BCIs, the calibration problem remains as one of the biggest obstacles for BCIs.

Section summary and References

Summary. In this section, we reviewed the challenges of calibrating BCIs and looked at a few approaches to address these challenges. In the next chapter, I will present my research on BCIs.

References. Lotte discusses in his paper Lotte (2015) different approaches to minimize or suppress calibration time for BCIs.

7

Contributions

As stated in the introduction, the contributions of this thesis are twofold – firstly, a wide-ranging background in the first chapters, and secondly, the research I have done, which I present in this and the following chapter. The first section of this chapter briefly introduces the published papers that are included in the thesis, and the second section the papers that are not included. You find the included papers at the end of the thesis.

7.1 Included Papers

The included papers showcase my research on BCIs and are all to some extent related to the calibration of EEG-based BCI systems.

Paper I presents the theory of Multi-Armed Bandits and how Multi-Armed Bandits are and can be used in BCIs. In particular, the theory from Multi-Armed Bandits can be used during calibration to optimize data selection for transfer learning.

Paper II presents a framework for developing BCIs that, for example, can be used to test different algorithms for BCIs, such as calibration methods.

Paper III is an abstract for a poster session at a BCI conference that presents the idea of using Markov Decision Processes for an adaptive BCI that decides when to recalibrate the BCI for better performance and when to use the BCI system as it is. This idea evolved into the idea of using Multi-Armed Bandits for BCIs (Paper I).

Paper I

F. Heskebeck, C. Bergeling, and B. Bernhardsson (2022). "Multi-Armed Bandits in Brain-Computer Interfaces". *Frontiers in Human Neuroscience* **16**. ISSN: 1662-5161. DOI: 10.3389/fnhum.2022.931085

Summary. This paper presents the theory of multi-armed bandits. The basic concept for a multi-armed bandit problem is the following. If you are presented with multiple choices, also called actions, where each action gives you an unknown reward, the challenge is to choose the best action. Once you have taken an action, you get the reward, and you can then, based on your previously given rewards, guess what reward an action will give you. Then, the question is if you should exploit the action you think is best or explore other actions to see if they are any better. Multi-armed bandits are one of the basic Reinforcement Learning algorithms to tackle this exploration versus exploitation tradeoff.

After presenting the theory of multi-armed bandits, the paper shows some examples where multi-armed bandits are used in BCIs today and provides some examples of future uses.

Contribution. The conceptualization of the paper was done collectively by all authors. F. Heskebeck did the literature research and wrote the manuscript with input and revisions from all co-authors.

Paper II

M. Gemborn Nilsson, P. Tufvesson, F. Heskebeck, and M. Johansson (2023). "An open-source human-in-the-loop BCI research framework: method and design". *Frontiers in Human Neuroscience* **17**. ISSN: 1662-5161. DOI: 10.3389/fnhum.2023.1129362

Summary. This paper presents a framework for developing real-time BCIs. It also lists current tools for BCIs and elaborates on technical details for developing BCI frameworks. It showcases two applications, one based on the motor imagery paradigm and one on the P300 paradigm.

I was involved in developing the real-time processing and classification of EEG data with timeflux (called "Calculate program" in the paper).

Contribution. F. Heskebeck and M. Gemborn Nilsson initially developed the software for this paper. During F. Heskebeck's parental leave, M. Gemborn Nilsson and P. Tufvesson continued working on the project and wrote the paper. All authors helped with the final revision of the paper.

Paper III

F. Heskebeck and C. Bergeling (2021). "An Adaptive Approach for Task-Driven BCI Calibration". In: *BCI Meeting 2021*. BCI Meeting 2021. URL: http://lup.lub.lu.se/record/de71d9e1-3dfb-46d2-b54f-61abe48a8d2d

Summary. Over time, the EEG signals change, and the BCI system needs to be recalibrated. This paper is an abstract that presents the idea of using Markov Decision Processes to decide if the BCI should be used as it is or recalibrated for better performance.

Contribution. The conceptualization of the paper was done collectively by both authors. F. Heskebeck wrote the manuscript with input and revisions from the co-author.

7.2 Additional papers

I have also contributed to the following peer-reviewed paper. It is unrelated to BCIs and, therefore, not included in the thesis.

M. Bauer and F. Heskebeck (Jan. 2022). “A Workplace Equality Workshop for the Control Engineering Classroom”. *IFAC-PapersOnLine*. 13th IFAC Symposium on Advances in Control Education ACE 2022 **55**:17, pp. 97–102. ISSN: 2405-8963. DOI: 10.1016/j.ifacol.2022.09.231

8

Discussion

In the previous chapters, I have given you a broad background on BCIs, ranging from algorithms to ethics, and an overview of my contributions in the subfield of calibration of BCIs. In this chapter, I discuss my contributions in-depth and suggest future work.

The overall title for the research project I'm part of is "Realtime Individualization of BCIs – Optimizing the Next Generation BCIs using Cloud Computing". We have limited ourselves to EEG-based Brain-Computer Interfaces (BCIs) since they are the most commonly used. It lies in the project's scope to examine and develop state-of-the-art methods. My research started with generally learning about Brain-Computer Interfaces (BCIs), and I gained interest in the calibration part of the BCIs. I liked the calibration problem's data-driven nature while still focusing on the math behind the algorithms.

8.1 Insights

In this section, I discuss my insights from the papers. I start with paper III since it is my first paper in chronological order. Then, continue with paper I, which builds upon the ideas from paper III. Finally, I discuss paper II, which is separate from the other two.

Paper III

The problem formulation for Paper III was an adaptive BCI system that should decide if it should be used as it is or recalibrated for better performance. We formulated the decision-making as a Markov decision process [Littman, 2001]. In a Markov decision process, the system can be in different states and transition after an action to another (or the same) state with a given probability. We formulated a simple situation for our BCI case and derived theoretical results that told us the conditions for when to use the BCI as it is and when to recalibrate. The problem with our formulation was that it is hard to implement in practice. The state and transition probabilities were more or less impossible to identify in a real-world setting, so we concluded that while the idea with an adaptive BCI was good, we needed another approach than the Markov decision process formulation.

Paper I

With an adaptive decision-making BCI in mind, we considered multi-armed bandits for modeling the decision-making. Inspired by Fruitet et al. (2012), we wanted to use multi-armed bandits for data selection during calibration. When working with BCIs the classification is rarely perfect and can often be improved by providing more data to the underlying machine learning algorithm – remember, the aim during calibration of BCIs is to train a machine learning algorithm. Instead of sampling data from all possible classes, the challenge is identifying which data category will improve classification the most. We framed this challenge as a multi-armed bandit problem. One challenge with this formulation was how rewards should be given after an action is taken – should it be the classification accuracy, a binary reward for if the classification was improved, or something else entirely? Another challenge was the nonstationarity of the problem, which made it hard to study – adding more data to the training set will change how all classes are classified, not only the classification of the new data's class. To get more insight into how to handle these challenges, I did an extensive literature review on Multi-armed bandits, which ended up as a review paper (Paper I).

While writing the review paper, I got new inspiration for how to use multi-armed bandits in BCIs, this time from Gutiérrez et al. (2017). The new idea was to use multi-armed bandits for data selection for transfer learning. The idea was for the BCI to choose data samples from different sources (other users or previous sessions) to be used for transfer learning for this session. We formulated this as a multi-armed bandit problem where picking data from a source was an action, and the classification accuracy when using that data for transfer learning was the corresponding reward. Preliminary results from this showed that the best approach often was to pick data samples from the same source all the time rather than exploring other sources. I still believe that this is a promising approach, but while studying this, I observed that even though transfer learning is used, using data from some subjects results in better performance than using data for other persons. It made me curious, and I gained interest in Riemann geometry-based transfer learning methods, which brings us to my plans for future work (see Section 8.2).

Paper II

My main insight from working on Paper II was that designing a BCI is hard. There are so many things that need to sync for it to work at all, and even when everything is in sync, you still need to have a good experiment and use the EEG hardware correctly for the signals to be of sufficient quality. This motivates me to use publicly available EEG data for future research since that data is "correctly" measured and collected.

A second insight from working on Paper II was all the different algorithms that exist for BCIs. In this thesis, I have presented the algorithms I most frequently encountered in the literature, but there are more.

Yet another insight is that I realized how BCIs connect many different research fields. You need to have an understanding of how the brain works to be able to design meaningful experiments, you need to be a hardcore programmer to implement anything that is not a standard BCI, and you need to be a mathematician to understand how the used algorithms work. To reach the next level of BCIs, experts in all these areas need to work together.

8.2 Future work

There is a lot of exciting research that can be done on BCIs. Here, I present some ideas I will study hereafter, and some research areas I think will improve BCIs significantly.

For me

As stated above, the research project I'm part of aims to explore and develop state-of-the-art methods for BCIs. I have recently gained interest in Riemannian geometry-based classifiers and transfer learning, which are state-of-the-art methods. The paper by Rodrigues et al. (2019) sparked my interest, which describes a transfer learning method based on Riemannian geometry. Following this inspiration, I plan to investigate the following two topics:

- **Data for transfer learning** – When we do transfer learning, we use source data from a previous session, another user, or a similar task to train the machine learning algorithm for a new session (target data). My preliminary results show that when using data from some persons, the BCI performance is better than if data from other persons is used, even though transfer learning is done. I'm interested in exploring if there are any features of the data that can predict what source data is suitable to use with the target data.

- **Recalibration of BCI systems** – I still believe in the idea of an automated BCI system that can detect when recalibration is needed (same idea as in Paper III). I'm interested in creating such as automatic BCI using algorithms based on Riemannian geometry.

For everyone

In my thesis, I have focused on the calibration challenge of BCIs. But many other challenges with BCIs must be solved before we see BCIs in our everyday lives. Here are three research areas that I believe will improve BCIs significantly.

- **Fundamental research on the brain** – With more knowledge of how the brain works, it would be easier to decipher the meaning of the measured brain signals. A lot of cool research is done in this area, but even more is needed.
- **Hardware** – Development of hardware that is easier and faster to use and gives better signal quality is needed.
- **New algorithms** – Generally, new algorithms and methods to process and analyze EEG data are needed.

Summary

This chapter discussed my insights from my papers and my plans for future work.

9

Final words

To summarize this thesis, we can conclude that Brain-Computer Interfaces are a technology of the future, and more research is needed before we reach that future.

In the thesis, we have learned what a BCI is and that the calibration of BCIs is one of the biggest challenges with BCIs. We have also learned about the brain and what brain signals are used in BCIs. We looked at many algorithms used in BCIs and saw that machine learning is at the heart of BCIs. In my research, I have investigated Markov decision processes and Multi-armed bandits as algorithms for automatic BCIs, as well as developed a framework for BCI design.

My contributions in this thesis are twofold: the first chapters give a wide-ranging background to Brain-Computer Interfaces, and the later chapters present my initial research on Brain-Computer Interfaces. In the future, I plan to delve into the realm of Riemannian geometry to improve the calibration of Brain-Computer Interfaces.

Thank you for reading my thesis!

FRIDA HESKEBECK

Figure credits

This chapter provides copyright information about all figures in the thesis (not including figures in the papers).

Figure 2.1	Running a BCI © 2022 by Frida Heskebeck is licensed under CC BY-SA 4.0.
Figure 3.1	Neuron © 2022 by Frida Heskebeck is licensed under CC BY-SA 4.0. Derived from Neuron © 2019 by Dhp1080 (CC BY-SA 3.0).
Figure 3.2	Medical gallery of Blausen Medical 2014 © 2014 by Blausen.com staff (CC BY 3.0).
Figure 3.3	Sensory Homunculus © 2013 by OpenStax College (CC BY 3.0).
Figure 3.4	Components of ERP © 2008 by Choms (original) and Mononomic (vectorized) (CC BY-SA 3.0).
Figure 3.5	Neuroimaging comparision © 2022 by Frida Heskebeck is licensed under CC BY-SA 4.0. Inspired by Nam et al. (2018).
Figure 3.6	© 2019 by Kennet Ruona (free use within Lund University).
Figure 3.7	EEG 10-10 system with additional information (CC0 1.0).
Figure 3.8	Human EEG with prominent alpha-rhythm © 2015 by Andrii Cherninskyi (CC BY-SA 4.0).
Figure 3.9	EEG brainwaves (CC0 1.0).
Figure 3.10	NIMH MEG (public domain).
Figure 3.11	1206 FMRI © 2016 by OpenStax (CC BY 4.0).
Figure 3.12	Blonde fNIRS lady © 2019 by Elisenicolegray (CC BY-SA 4.0).
Figure 3.13	ECAT-Exact-HR–PET-Scanner (public domain).

Figure 4.1	Decoding flowchart © 2023 by Frida Heskebeck is licensed under CC BY-SA 4.0.
Figure 4.2	Human EEG artefacts © 2015 by Andrii Cherninskyi (CC BY-SA 4.0).
Figure 4.3	Power Spectral Density, generated with the MNE toolbox in Python. Same data as in this demo (https://mne.tools/stable/auto_examples/decoding/decoding_csp_eeg.html) and using the `Raw.compute_psd` function.
Figure 4.4	Time-Frequency plot, generated with the MNE toolbox in Python. Same data as in this demo (https://mne.tools/stable/auto_examples/decoding/decoding_csp_eeg.html) and using the `time_frequency.tfr_morlet` function.
Figure 4.5	Illustrative PCA plots © 2023 by Frida Heskebeck is licensed under CC BY-SA 4.0.
Figure 4.6	Illustrative CSP plots © 2023 by Frida Heskebeck is licensed under CC BY-SA 4.0.
Figure 4.7	CSP topo-maps, generated with the MNE toolbox in Python. Demo from MNE: https://mne.tools/stable/auto_examples/decoding/decoding_csp_eeg.html.
Figure 4.8	Illustrative MDM and LDA plots © 2023 by Frida Heskebeck is licensed under CC BY-SA 4.0.
Figure 4.9	Illustrative SVM plots © 2023 by Frida Heskebeck is licensed under CC BY-SA 4.0.
Figure 4.10	Illustration of neural network structure © 2023 by Frida Heskebeck is licensed under CC BY-SA 4.0.
Figure 4.11	Comparison of elliptic, Euclidean and hyperbolic geometries in two dimensions © 2020 by CMG Lee (CC BY-SA 4.0).

Bibliography

Abiri, R., S. Borhani, E. W. Sellers, Y. Jiang, and X. Zhao (2019). “A comprehensive review of EEG-based brain-computer interface paradigms”. *Journal of Neural Engineering* **16**:1, p. 011001. ISSN: 1741-2552. DOI: 10.1088/1741-2552/aaf12e. pmid: 30523919.

Aubé, T. (20, 2019). *The Most Important Tech No One Is Talking About*. HackerNoon.com. URL: https://medium.com/hackernoon/the-most-important-tech-no-one-is-talking-about-c171c438e1a (visited on 2021-09-08).

Barachant, A., S. Bonnet, M. Congedo, and C. Jutten (2012). “Multiclass Brain–Computer Interface Classification by Riemannian Geometry”. *IEEE Transactions on Biomedical Engineering* **59**:4, pp. 920–928. ISSN: 1558-2531. DOI: 10.1109/TBME.2011.2172210.

Bauer, M. and F. Heskebeck (2022). “A Workplace Equality Workshop for the Control Engineering Classroom”. *IFAC-PapersOnLine*. 13th IFAC Symposium on Advances in Control Education ACE 2022 **55**:17, pp. 97–102. ISSN: 2405-8963. DOI: 10.1016/j.ifacol.2022.09.231.

Berger, H. (1929). “Uber das elektroenkephalogramm des menschen”. *Archiv für psychiatrie und nervenkrankheiten* **87**, pp. 527–570.

Bhatt, D., C. Patel, H. Talsania, J. Patel, R. Vaghela, S. Pandya, K. Modi, and H. Ghayvat (2021). “CNN Variants for Computer Vision: History, Architecture, Application, Challenges and Future Scope”. *Electronics* **10**:20, p. 2470. ISSN: 2079-9292. DOI: 10.3390/electronics10202470. (Visited on 2023-07-10).

Blankertz, B. (26, 2018). “Gentle Introduction to Signal Processing and Classification for Single-Trial EEG Analysis”. In: Nam, C. S. et al. (Eds.). *Brain–Computer Interfaces Handbook: Technological and Theoretical Advances*. CRC Press, New York, pp. 343–370. ISBN: 978-1-351-23195-4. DOI: 10.1201/9781351231954.

Blankertz, B., R. Tomioka, S. Lemm, M. Kawanabe, and K.-r. Muller (2008). "Optimizing Spatial filters for Robust EEG Single-Trial Analysis". *IEEE Signal Processing Magazine* **25**:1, pp. 41–56. ISSN: 1558-0792. DOI: 10.1109/MSP.2008.4408441.

Burwell, S., M. Sample, and E. Racine (9, 2017). "Ethical aspects of brain computer interfaces: a scoping review". *BMC Medical Ethics* **18**:1, p. 60. ISSN: 1472-6939. DOI: 10.1186/s12910-017-0220-y. URL: https://doi.org/10.1186/s12910-017-0220-y (visited on 2022-02-04).

eigenchris (2017). *Tensors For Beginners*. [Video]. URL: https://www.youtube.com/playlist?list=PLJHszsWbB6hrkmmq57lX8BV-o-YIOFsiG (visited on 2023-08-22).

eigenchris (2020). *Tensor Calculus*. [Video]. URL: https://www.youtube.com/playlist?list=PLJHszsWbB6hpk5h8lSfBkVrpjsqvUGTCx (visited on 2023-08-22).

Facebook (14, 2021). *BCI milestone: New research from UCSF with support from Facebook shows the potential of brain-computer interfaces for restoring speech communication*. Facebook Technology. URL: https://tech.fb.com/bci-milestone-new-research-from-ucsf-with-support-from-facebook-shows-the-potential-of-brain-computer-interfaces-for-restoring-speech-communication/ (visited on 2021-09-08).

Faisal, A. A., M. P. Deisenroth, and C. S. Ong (2020a). "Dimensionality Reduction with Principal Component Analysis". In: *Mathematics for Machine Learning*. Cambridge University Press, pp. 317–347. URL: https://mml-book.com/ (visited on 2021-06-15).

Faisal, A. A., M. P. Deisenroth, and C. S. Ong (2020b). "Probability and Distributions". In: *Mathematics for Machine Learning*. Cambridge University Press, pp. 172–224. URL: https://mml-book.com/ (visited on 2021-06-15).

Farwell, L. A. and E. Donchin (1, 1988). "Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials". *Electroencephalography and Clinical Neurophysiology* **70**:6, pp. 510–523. ISSN: 0013-4694. DOI: 10.1016/0013-4694(88)90149-6. URL: https://www.sciencedirect.com/science/article/pii/0013469488901496 (visited on 2021-09-11).

Fruitet, J., A. Carpentier, R. Munos, and M. Clerc (2012). "Bandit Algorithms boost Brain Computer Interfaces for motor-task selection of a brain-controlled button". In: Bartlett, P. et al. (Eds.). *Advances in Neural Information Processing Systems*. Vol. 25. Neural Information Processing Systems (NIPS) Foundation, Lake Tahoe, Nevada, United States, pp. 458–466. URL: https://hal.archives-ouvertes.fr/hal-00771495 (visited on 2021-10-29).

Gemborn Nilsson, M., P. Tufvesson, F. Heskebeck, and M. Johansson (2023). "An open-source human-in-the-loop BCI research framework: method and design".

Frontiers in Human Neuroscience **17**. ISSN: 1662-5161. DOI: 10.3389/fnhum.2023.1129362.

Géron, A. (5, 2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. "O'Reilly Media, Inc.". 851 pp. ISBN: 978-1-4920-3261-8. Google Books: HHetDwAAQBAJ.

Gonfalonieri, A. (6, 2020). "What Brain-Computer Interfaces Could Mean for the Future of Work". *Harvard Business Review*. ISSN: 0017-8012. URL: https://hbr.org/2020/10/what-brain-computer-interfaces-could-mean-for-the-future-of-work (visited on 2021-09-11).

Google (2021). *Google Brain Team*. URL: https://research.google/teams/brain/ (visited on 2021-09-08).

Gutiérrez, B., L. Peter, T. Klein, and C. Wachinger (2017). "A Multi-armed Bandit to Smartly Select a Training Set from Big Medical Data". In: Descoteaux, M. et al. (Eds.). *Medical Image Computing and Computer Assisted Intervention - MICCAI 2017*. Lecture Notes in Computer Science. Springer International Publishing, Cham, pp. 38–45. ISBN: 978-3-319-66179-7. DOI: 10.1007/978-3-319-66179-7_5.

Heskebeck, F. and C. Bergeling (2021). "An Adaptive Approach for Task-Driven BCI Calibration". In: *BCI Meeting 2021*. BCI Meeting 2021. URL: http://lup.lub.lu.se/record/de71d9e1-3dfb-46d2-b54f-61abe48a8d2d.

Heskebeck, F., C. Bergeling, and B. Bernhardsson (2022). "Multi-Armed Bandits in Brain-Computer Interfaces". *Frontiers in Human Neuroscience* **16**. ISSN: 1662-5161. DOI: 10.3389/fnhum.2022.931085.

Jayaram, V. and A. Barachant (2018). "MOABB: trustworthy algorithm benchmarking for BCIs". *Journal of Neural Engineering* **15**:6, p. 066011. ISSN: 1741-2552. DOI: 10.1088/1741-2552/aadea0. URL: https://doi.org/10.1088/1741-2552/aadea0 (visited on 2021-09-02).

Jayaram, V., K.-H. Fiebig, J. Peters, and M. Grosse-Wentrup (26, 2018). "Transfer Learning for BCIs". In: Nam, C. S. et al. (Eds.). *Brain–Computer Interfaces Handbook: Technological and Theoretical Advances*. CRC Press, New York, pp. 425–441. ISBN: 978-1-351-23195-4. DOI: 10.1201/9781351231954.

King, J.-R. and V. Wyart (2, 2021). "The Human Brain Encodes a Chronicle of Visual Events at each Instant of Time thanks to the Multiplexing of Traveling Waves". *Journal of Neuroscience*. ISSN: 0270-6474, 1529-2401. DOI: 10.1523/JNEUROSCI.2098-20.2021. pmid: 33811150. URL: https://www.jneurosci.org/content/early/2021/04/01/JNEUROSCI.2098-20.2021 (visited on 2021-08-13).

Krol, L. R., L. M. Andreesen, and T. O. Zander (26, 2018). "Passive Brain-Computer Interfaces: A Perspective on Increased Interactivity". In: Nam, C. S. et al. (Eds.). *Brain–Computer Interfaces Handbook: Technological and Theo-*

retical Advances. CRC Press, New York, pp. 70–86. ISBN: 978-1-351-23195-4. DOI: 10.1201/9781351231954.

Littman, M. L. (2001). “Markov Decision Processes”. In: Smelser, N. J. et al. (Eds.). *International Encyclopedia of the Social & Behavioral Sciences*. Pergamon, Oxford, pp. 9240–9242. ISBN: 978-0-08-043076-8. DOI: 10.1016/B0-08-043076-7/00614-8. (Visited on 2023-08-29).

Lotte, F., L. Bougrain, A. Cichocki, M. Clerc, M. Congedo, A. Rakotomamonjy, and F. Yger (2018a). “A review of classification algorithms for EEG-based brain-computer interfaces: a 10 year update”. *Journal of Neural Engineering* **15**:3, p. 031005. ISSN: 1741-2552. DOI: 10.1088/1741-2552/aab2f2. pmid: 29488902.

Lotte, F., M. Congedo, A. Lécuyer, F. Lamarche, and B. Arnaldi (2007). “A review of classification algorithms for EEG-based brain-computer interfaces”. *Journal of Neural Engineering* **4**:2, R1–R13. ISSN: 1741-2560. DOI: 10.1088/1741-2560/4/2/R01. pmid: 17409472.

Lotte, F. (2015). “Signal Processing Approaches to Minimize or Suppress Calibration Time in Oscillatory Activity-Based Brain–Computer Interfaces”. *Proceedings of the IEEE* **103**:6, pp. 871–890. ISSN: 1558-2256. DOI: 10.1109/JPROC.2015.2404941.

Lotte, F., C. S. Nam, and A. Nijholt (26, 2018b). “Introduction: Evolution of Brain–Computer Interfaces”. In: Nam, C. S. et al. (Eds.). *Brain–Computer Interfaces Handbook: Technological and Theoretical Advances*. CRC Press, New York, pp. 1–8. ISBN: 978-1-351-23195-4. DOI: 10.1201/9781351231954.

Marzbani, H., H. R. Marateb, and M. Mansourian (2016). “Neurofeedback: A Comprehensive Review on System Design, Methodology and Clinical Applications”. *Basic and Clinical Neuroscience* **7**:2, pp. 143–158. ISSN: 2008-126X. DOI: 10.15412/J.BCN.03070208. pmid: 27303609.

Microsoft (2021). *Brain-Computer Interfaces*. Microsoft Research. URL: https://www.microsoft.com/en-us/research/project/brain-computer-interfaces/ (visited on 2021-09-08).

Müller-Putz, G. R. (1, 2020). “Chapter 18 - Electroencephalography”. In: Ramsey, N. F. et al. (Eds.). *Handbook of Clinical Neurology*. Vol. 168. Brain-Computer Interfaces. Elsevier, pp. 249–262. DOI: 10.1016/B978-0-444-63934-9.00018-4. URL: https://www.sciencedirect.com/science/article/pii/B9780444639349000184 (visited on 2022-01-20).

Muratore, D. G. and E. J. Chichilnisky (2020). “Artificial Retina: A Future Cellular-Resolution Brain-Machine Interface”. In: Murmann, B. et al. (Eds.). *NANO-CHIPS 2030: On-Chip AI for an Efficient Data-Driven World*. The Frontiers Collection. Springer International Publishing, Cham, pp. 443–465. ISBN: 978-3-030-18338-7. DOI: 10.1007/978-3-030-18338-7_24. URL: https://doi.org/10.1007/978-3-030-18338-7_24 (visited on 2021-09-08).

Nam, C. S., I. Choi, A. Wadeson, and M. Whang (26, 2018). "Brain–Computer Interface: An Emerging Interaction Technology". In: Nam, C. S. et al. (Eds.). *Brain–Computer Interfaces Handbook: Technological and Theoretical Advances*. CRC Press, New York, pp. 12–52. ISBN: 978-1-351-23195-4. DOI: 10.1201/9781351231954.

National Institute of Neurological Disorders and Stroke (2023). *Brain Basics: Know Your Brain*. https://www.ninds.nih.gov/health-information/public-education/brain-basics/brain-basics-know-your-brain. (Visited on 2023-09-14).

Neuralink (2021). *Neuralink*. Neuralink. URL: https://neuralink.com/ (visited on 2021-09-08).

Nicolas-Alonso, L. F. and J. Gomez-Gil (31, 2012). "Brain Computer Interfaces, a Review". *Sensors (Basel, Switzerland)* **12**:2, pp. 1211–1279. ISSN: 1424-8220. DOI: 10.3390/s120201211. pmid: 22438708. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3304110/ (visited on 2021-08-13).

Nijboer, F. and U. Broermann (2010). "Brain–Computer Interfaces for Communication and Control in Locked-in Patients". In: Graimann, B. et al. (Eds.). *Brain-Computer Interfaces: Revolutionizing Human-Computer Interaction*. The Frontiers Collection. Springer, Berlin, Heidelberg, pp. 185–201. ISBN: 978-3-642-02091-9. DOI: 10.1007/978-3-642-02091-9_11. URL: https://doi.org/10.1007/978-3-642-02091-9_11 (visited on 2022-01-19).

Pfurtscheller, G. and C. Neuper (2010). "Dynamics of Sensorimotor Oscillations in a Motor Task". In: Graimann, B. et al. (Eds.). *Brain-Computer Interfaces: Revolutionizing Human-Computer Interaction*. The Frontiers Collection. Springer, Berlin, Heidelberg, pp. 47–64. ISBN: 978-3-642-02091-9. DOI: 10.1007/978-3-642-02091-9_3. URL: https://doi.org/10.1007/978-3-642-02091-9_3 (visited on 2022-01-19).

Purves, D., G. J. Augustine, D. Fitzpatrick, W. C. Hall, A.-S. LaMantia, R. D. Mooney, M. L. Platt, and L. E. White, (Eds.) (2018). *Neuroscience*. 6th edition. Oxford University Press, New York. 960 pp. ISBN: 978-1-60535-380-7.

Riggins, T. and L. S. Scott (2020). "P300 development from infancy to adolescence". *Psychophysiology* **57**:7, e13346. ISSN: 1469-8986. DOI: 10.1111/psyp.13346. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/psyp.13346 (visited on 2022-02-08).

Rodrigues, P. L. C., C. Jutten, and M. Congedo (2019). "Riemannian Procrustes Analysis: Transfer Learning for Brain–Computer Interfaces". *IEEE Transactions on Biomedical Engineering* **66**:8, pp. 2390–2401. ISSN: 1558-2531. DOI: 10.1109/TBME.2018.2889705.

Ruiz-Blondet, M. V., Z. Jin, and S. Laszlo (2016). "CEREBRE: A Novel Method for Very High Accuracy Event-Related Potential Biometric Identification". *IEEE Transactions on Information Forensics and Security* **11**:7, pp. 1618–1629. ISSN: 1556-6021. DOI: 10.1109/TIFS.2016.2543524.

Saha, S., K. A. Mamun, K. Ahmed, R. Mostafa, G. R. Naik, S. Darvishi, A. H. Khandoker, and M. Baumert (2021). "Progress in Brain Computer Interface: Challenges and Opportunities". *Frontiers in Systems Neuroscience* **15**, p. 4. ISSN: 1662-5137. DOI: 10.3389/fnsys.2021.578875. URL: https://www.frontiersin.org/article/10.3389/fnsys.2021.578875 (visited on 2021-09-11).

scikit-learn developers (2023). *Scikit-learn: User guide*. https://scikit-learn/stable/user_guide.html. (Visited on 2023-04-25).

Sur, S. and V. K. Sinha (2009). "Event-related potential: An overview". *Industrial Psychiatry Journal* **18**:1, pp. 70–73. ISSN: 0972-6748. DOI: 10.4103/0972-6748.57865. pmid: 21234168. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3016705/ (visited on 2022-01-17).

This Is CYBATHLON (2022). URL: https://cybathlon.ethz.ch/en/cybathlon (visited on 2022-01-14).

Vialatte, F.-B., M. Maurice, J. Dauwels, and A. Cichocki (1, 2010). "Steady-state visually evoked potentials: Focus on essential paradigms and future perspectives". *Progress in Neurobiology* **90**:4, pp. 418–438. ISSN: 0301-0082. DOI: 10.1016/j.pneurobio.2009.11.005. URL: https://www.sciencedirect.com/science/article/pii/S0301008209001853 (visited on 2022-02-22).

Vidal, J. J. (1973). "Toward Direct Brain-Computer Communication". *Annual Review of Biophysics and Bioengineering* **2**:1, pp. 157–180. ISSN: 0084-6589, 0084-6589. DOI: 10.1146/annurev.bb.02.060173.001105. URL: http://www.annualreviews.org/doi/abs/10.1146/annurev.bb.02.060173.001105 (visited on 2021-09-11).

Printed by Media-Tryck, Lund 2023 NORDIC SWAN ECOLABEL 3041 0903

Lund University

Department of Automatic Control
P.O. Box 118, 221 00 Lund, Sweden
www.control.lth.se

ISRN LUTFD2/TFRT-3281
ISSN 0280–5316