UNIVERSITY OF TARTU
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
Institute of Computer Science
Computer Science Curriculum

Anti Ingel

Control a Robot via VEP Using Emotiv EPOC

Bachelor's Thesis (9 ECTS)

Supervisor: Ilya Kuzovkin, MSc

Tartu 2015

Control a Robot via VEP Using Emotiv EPOC

Abstract

This thesis describes an SSVEP-based BCI implemented as a practical part of this work. One possible usage of a BCI that efficiently implements a communication channel between the brain and an external device would be to help severely disabled people to control devices that currently require pushing buttons, for example an electric wheelchair. The BCI implemented as a part of this thesis uses widely known PSDA and CCA feature extraction methods and introduces a new way to combine these methods. Combining different methods improves the performance of a BCI. The application was tested only superficially and the following results were obtained: 2.61 ± 0.28 s target detection time, 85.81 ± 6.39 % accuracy and 27.73 ± 5.11 bits/min ITR. The implemented BCI is open-source, written in Python 2.7, has graphical user interface and uses inexpensive EEG device called Emotiv EPOC. The BCI requires only a computer and Emotiv EPOC, no additional hardware is needed. Different EEG devices could be used after modifying the code.

Keywords

electroencephalography (EEG), brain-computer interface (BCI), steady-state visual evoked potential (SSVEP), canonical correlation analysis (CCA), power spectrum density analysis (PSDA), open-source

Visuaalse stiimuliga esilekutsutud potentsiaalidel põhinev roboti juhtimine Emotiv EPOC seadmega

Lühikokkuvõte

Antud töö kirjeldab visuaalse stiimuliga esilekutsutud potentsiaalidel põhinevat aju ning arvuti vahelist liidest (AAL), mis loodi antud töö praktilise osana. AALi saab kasutada aju ja seadme vahelise otsese suhtluskanali loomiseks, mis tähendab, et seadmega suhtlemiseks pole vaja nuppe vajutada, piisab vaid visuaalsete stiimulite vaatamisest. Efektiivne AAL võimaldaks raske puudega isikutel näiteks elektroonilist ratastooli juhtida. Antud töö osana loodud AAL kasutab tuntud kanoonilise korrelatsiooni- ja võimsusspektri analüüsi meetodeid ning uuendusena kombineerib need kaks meetodit üheks teineteist täiendavaks meetodiks. Kahe meetodi kombinatsioon muudab AALi täpsemaks. AALi testiti antud töös vaid pealiskaudselt ning tulemused on järgnevad: ühe käsu edastamise aeg 2.61 ± 0.28 s, täpsus 85.81 ± 6.39 % ning informatsiooni edastamise kiirus 27.73 ± 5.11 bitt/min. Antud AAL on avatud lähtekoodiga, kirjutatud Python 2.7 programmeerimiskeeles, sisaldab graafilist kasutajaliidest ning kasutab aju tegevuse mõõtmiseks elektroensefalograafia (EEG) seadet Emotiv EPOC. AALi kasutamiseks on vaja ainult arvutit ja Emotiv EPOC seadet. Koodi muutes on võimalik kasutada ka teisi EEG seadmeid.

Võtmesõnad

Elektroensefalograafia EEG, aju-arvuti liides (AAL), visuaalne stiimul, kanooniline korrelatisoonianalüüs, võimsusspektri analüüs, avatud lähtekood

Table of contents

Introduction

Over the last decade the question how to implement direct communication channel between the brain and an external device has received much attention [16, 28, 40, 45, 49, 50]. This communication channel works by translating recorded brain activity into commands to control the device or the other way round—by sending signals into the brain. Sending signals into the brain requires a surgery to implant the device that can send those signals. This approach can be used for example to repair damaged vision or hearing.

Measuring the brain activity, however, can be done without implanting devices into the brain and therefore it is easier and less expensive than invasive methods. This approach can be used for example to operate a wheelchair or for typing without having to press buttons. Nowadays the brain activity can be measured using non-invasive and inexpensive devices and thus many people could benefit from an application that efficiently implements this new communication channel between the brain and an external device.

This thesis describes an application that uses an electroencephalography (EEG) device called Emotiv EPOC to measure brain activity and translates the recorded signal into commands to control a robot. The application was written as a practical part of this thesis and the focus was on implementing an inexpensive application. Compared to existing applications, the practical part of this thesis combines two widely used methods called power spectrum density analysis (PSDA) [12] and canonical correlation analysis (CCA) [28] for extracting information from brain recording in a way which to the best of the author's knowledge has not been done before.

The first chapter of this thesis describes the biology of the brain and discusses, what aspect of the brain activity can actually be measured. In this chapter also different techniques and devices that can be used to measure brain activity are compared.

The second chapter describes how to evoke a brain response that can be extracted from the recording of brain activity using only Emotiv EPOC and a laptop computer. The methods that can be used to analyse the recording are discussed in this chapter.

The third chapter describes the application implemented as a practical part of this thesis. This chapter contains overview of related works, describes the signal processing and explains the novelty of the application.

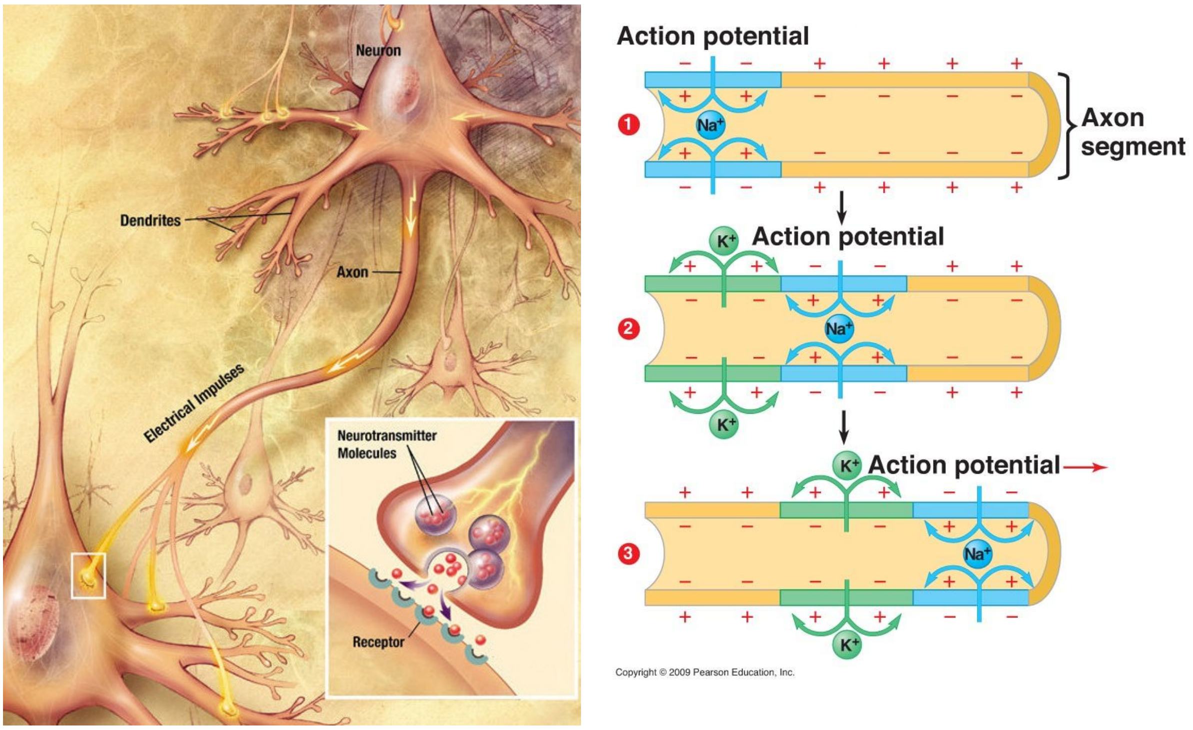
1 Electrical activity in the brain

The aim of this chapter is to describe the biology of the brain, discuss how brain activity can be measured and where the measurable activity originates from. In this chapter also different techniques and devices used to measure brain activity are compared and finally suitable device for controlling a robot is chosen.

1.1 Source of the electrical activity

As all living organisms are composed of cells, so are humans and the human brain. The brain consists of nerve cells called *neurons* and non-neural cells. There are approximately 86 billion neurons in the human brain and roughly as much non-neural cells [2]. The aim of this section is to describe how neurons interact with each other and discuss what aspect of this communication can be measured.

A typical neuron has a cell body, multiple nerve endings or *dendrites*, and one nerve fibre or *axon*. Both dendrites and axons can branch multiple times. Neurons interact with each other via electro-chemical signals that are transmitted through various connections. These connections are not static and can change over time. The connection between an axon and a dendrite is called a *synapse*. See figure 1.1a for an example of neurons and a synapse. The general rule is that a neuron sends signals through its axon and receives



(a) Neurons and a chemical synapse [41, p. 17].

(b) Action potential [10].

Figure 1.1: The structure of a neuron.

signals through dendrites. Functionally related neurons are connected to each other and form *neural pathways* [23].

To send signals, neurons must be able to maintain electric potential called *membrane potential*. Membrane potential of a neuron is the difference in electric potential between the inside of the neuron and the extracellular fluid around the neuron. When a neuron is not sending signals or in other words, when neuron is at a *resting state*, its membrane potential is slightly negative. The membrane potential of a neuron at a resting state is called *resting potential*. Negative resting potential is achieved by having more positively charged ions around the cell than inside the cell.

By having stable resting potential, neurons are able to send signals by rapidly increasing and decreasing the membrane potential along an axon. This event is called an *action potential*. See figure 1.1b for example. To increase or decrease the membrane potential of a cell, *ionophoric proteins* are used. These proteins transport ions across the cell membrane to regulate the concentration of ions inside the cell. Since ions are electrically charged, the concentration of ions inside a cell affects the membrane potential of the cell. The membrane potential of a cell can be increased by transporting positive ions into the cell and decreased by transporting positive ions out of the cell.

An action potential or signal sending is initiated when the membrane potential of a neuron exceeds certain threshold value called *threshold potential*. The membrane potential of a neuron can increase when the neuron receives signals from other neurons. The neuron that receives a signal is called *postsynaptic cell*. The received signal can cause the membrane potential of the postsynaptic cell to increase or decrease. This change in membrane potential is called a *postsynaptic potential*. If the postsynaptic potential is large enough for the membrane potential to exceed threshold potential, an action potential is initiated in the postsynaptic cell.

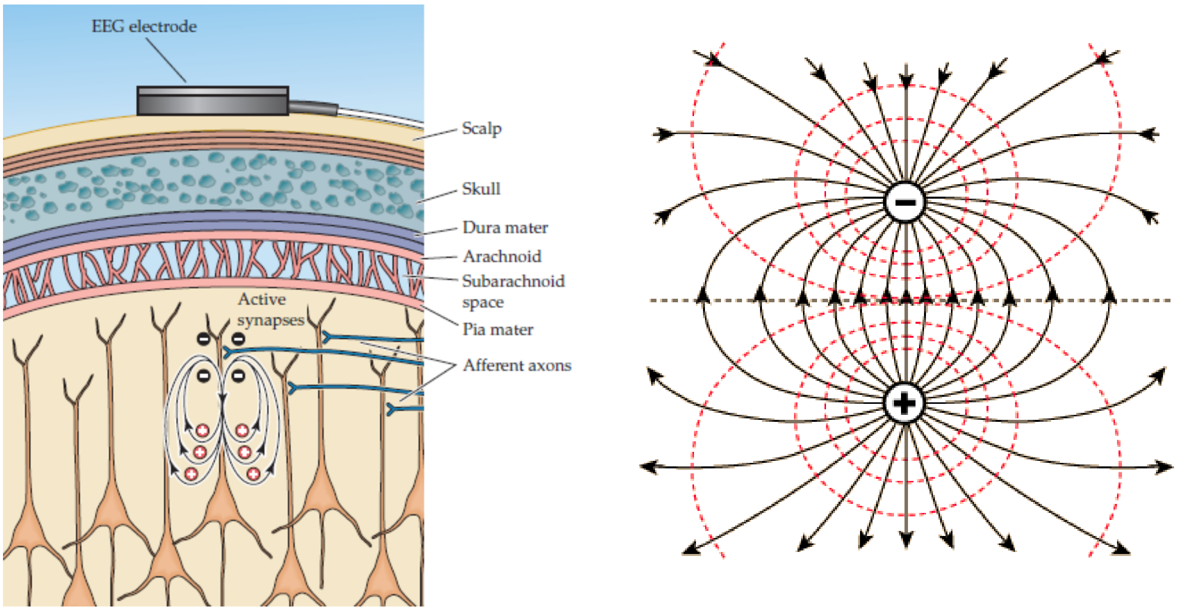
The following paragraph is mainly based on the article by Buzsaki *et al.* [9]. The postsynaptic potential is caused by ions flowing into the cell. To achieve electroneutrality, a balancing flow of ions from the interior to the exterior of the cell is needed. The ions of the balancing flow have the same electric charge as the entering ions. If the ions flowing into the cell have positive charge

- the membrane potential of the cell increases;
- location where positive ions enter the cell is called *sink*;
- location where positive ions exit the cell is called *source*; and
- as a result, there are less positive ions around sink and more around source;

If the ions flowing into the cell have negative charge

- the membrane potential of the cell decreases;
- location where negative ions enter the cell is called source;
- location where negative ions exit the cell is called sink; and
- as a result, there are less negative ions around source and more around sink.

In both cases, the source is more positively charged than the sink. Since the sink and the source have different electric potentials, they form a *current dipole*. See figure 1.2 for illustration.



(a) Current dipole generated by a neuron [37, p. 669].

(b) The electric field of an electric dipole[1].

Figure 1.2: Current dipole and electric dipole.

Current dipoles are important because it is possible to measure the electric field produced by these current dipoles from the scalp. See figure 1.2a for illustration. Measuring these electric fields is further discussed in section 1.3. Although action potentials generate stronger currents than postsynaptic potentials, their duration is short and nearby neurons rarely fire synchronously [9]. Thus recording the electrical activity of the brain from the scalp mainly relies on the electric fields of the current dipoles generated by neurons.

1.2 Overview of functional neuroimaging methods

As discussed in section 1.1, neurons in the brain are sending electrochemical signals to communicate with each other. There are several techniques available to measure this activity. The aim of this section is to briefly compare various non-invasive techniques. Measuring an aspect of brain function is called *functional neuroimaging* and common measurement methods divide into *haemodynamic* and *electromagnetic* techniques.

Haemodynamic techniques measure blood oxygenation and blood flow in the brain. More oxygen has to be delivered to more active brain regions and this allows the brain activity to be measured. Haemodynamic techniques include functional magnetic resonance imaging (fMRI), functional near-infrared spectroscopy (fNIRS) and positron emission tomography (PET).

Electromagnetic techniques measure either electrical activity or magnetic fields produced

[1] http://hyperphysics.phy-astr.gsu.edu/hbase/electric/equipot.html

by the electrical activity along the scalp. Electromagnetic techniques include EEG and magnetoencephalography (MEG). These methods have lower *temporal resolution* than haemodynamic methods, but measure only the activity in the outer layer of the brain. Temporal resolution shows how short time period can be reliably separated out by a measuring technique or in other words, how accurately the brain activity can be measured with respect to time.

To decide which method is best for controlling a robot, cost, portability and temporal resolution of each method is compared. See table 1.1 for details. For real-time robot controlling, lower temporal resolution is better because it enables faster decision making.

	Price from	Portable	Temporal resolution	Special requirements
MEG	millions[2]	no	milliseconds [13]	magnetically shielded room
fMRI	$150,000[3]	no	about 1 second [13]	magnetically shielded room
PET	$125,000[4]	no	about 1 second [13]	radioactive isotopes injection
fNIRS	$10,000 [38]	yes	over 0.1 second [38]	
EEG	$45[5]	yes	milliseconds [13]	

Table 1.1: Comparison of functional neuroimaging methods.

In this thesis, techniques that are portable and available to the consumer are preferred, so the usage would not be limited to certain location and would be available to the people in need. Considering all these arguments, it can be seen that currently the best choice for controlling a robot is an EEG device.

1.3 Overview of electroencephalography

As already mentioned in the previous section, EEG measures the electrical activity along the scalp. This electrical activity originates mainly from the electric fields generated by neurons as discussed in section 1.1. The aim of this section is to describe the basics of EEG and discuss how brain activity evoked by certain event can be measured using EEG.

The electric potential generated by one neuron is far too low to be recognized. Therefore, approximately 108 neurons have to have synchronous electrical activity to create a measurable field [33]. Furthermore, these neurons have to have certain orientation for the electric fields to add up and reach the electrode on the scalp. Namely, these neurons need to be perpendicular to the surface of the brain as shown in figure 1.2a.

EEG measures the potential fields as a function of voltage versus time using electrodes placed on scalp [33]. Since voltage is the electric potential difference between two points, one or more reference electrodes are commonly used. A voltmeter can be used to measure the difference in electric potential between two electrodes, one of which is the reference electrode. See figure 1.3 for a simplified scheme.

[2] http://neurogadget.com/2012/12/15/inexpensive-magnetoencephalography-meg-system-could-be-available-at-every-hospital/6495

[3] http://info.blockimaging.com/bid/92623/MRI-Machine-Cost-and-Price-Guide

[4] http://info.blockimaging.com/bid/68875/How-Much-Does-a-PET-CT-Scanner-Cost

[5] http://en.wikipedia.org/wiki/Comparison_of_consumer_brain-computer_interfaces

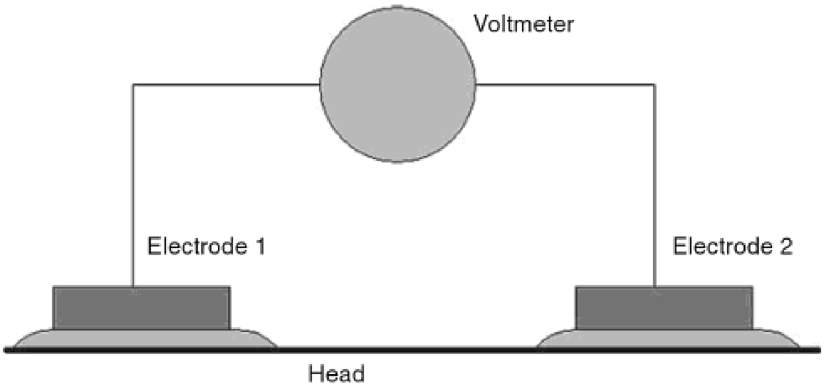


Figure 1.3: Electrodes connected to a voltmeter [26, p. 120].

Usually electrodes are placed on the scalp according to international 10-20 electrode location system. The outer layer of the brain can be classified into four lobes: temporal, occipital, parietal and frontal. See figure 1.4a for example. The mentioned 10-20 electrode location system uses a letter and a number to identify electrode location. The letter is the first letter of the brain lobe above which the electrode is located and therefore the electrode measures the activity of this brain lobe. There are more complicated electrode-naming-systems that extend the 10-20 system. See figure 1.6 for example.

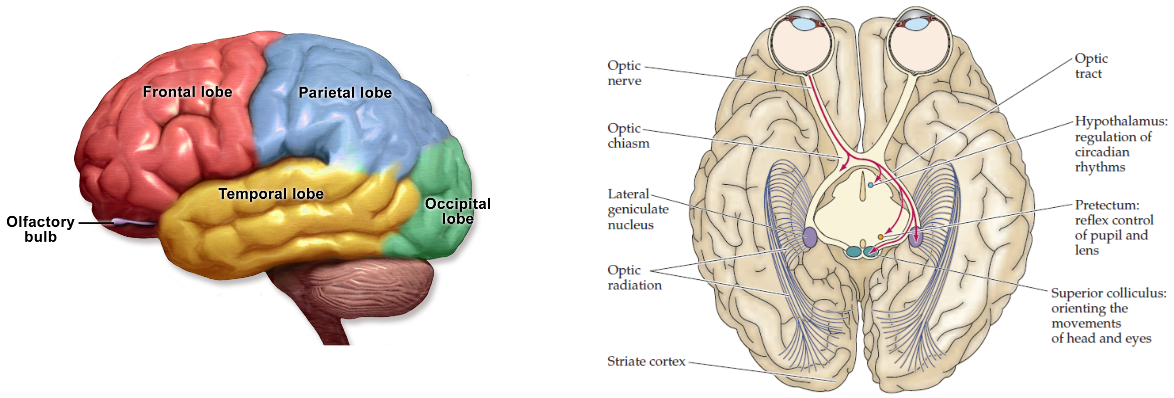
In a broad sense, EEG recording is linked to the general state of the brain [48]. Due to the generality of the recording, potentials evoked by certain events cannot be seen in the recording because the evoked potentials are much smaller than the general fluctuations. A brain potential evoked by some event is called event-related potential (ERP).

ERPs are linked to the information flow in the brain and are usually recorded by using an averaging technique [26]. For example, ERPs can be recorded by presenting a stimulus with a certain time interval to a subject and calculating the average of EEG signals recorded in the same time interval. This technique can be used for example if the stimulus is presented at a constant rate and ERPs are also evoked one after another in the same constant interval. As a result, ERPs are always evoked at certain time while other fluctuations and noise are random. Therefore, when calculating the average of the EEG recordings divided into the same time intervals as the stimulus presentation, the result will be an ERP because other fluctuations and noise mostly cancel out in the averaging process. Thus EEG can be used to record ERPs from the scalp.

1.4 Measuring visual evoked potentials

In section 1.3 brain potential called ERP was discussed and it was noted that ERPs are evoked by some events. The aim of this section is to describe an ERP called visual evoked potential (VEP) and discuss how VEPs can be measured. As the name suggests, VEPs are elicited by visual stimuli. The visual stimulus for eliciting a VEP can be very simple, for example a white square blinking on a black computer screen.

When the visual stimulus is seen, the signal travels from the eyes to the *visual processing centre* through the *primary visual pathway* in the brain [37]. The visual processing centre is located in the back of the brain. As mentioned in section 1.3, the outer layer of the brain can be classified into four lobes. The visual processing centre is located in the occipital lobe. See figure 1.4 for illustration. The primary visual pathway is a neural pathway; neural pathways were also mentioned in the section 1.1.


Frontal lobe
Parietal lobe
Temporal lobe
Occipital lobe
Olfactory bulb
Optic nerve
Optic chiasm
Lateral geniculate nucleus
Optic radiation
Striate cortex
Optic tract
Hypothalamus: regulation of circadian rhythms
Pretectum: reflex control of pupil and lens
Superior colliculus: orienting the movements of head and eyes

(a) Lobes of the brain [6]. (b) The primary visual pathway [37, p. 261].

Figure 1.4: The neural pathway from eyes to the occipital lobe.

Due to the posterior location of the occipital lobe, electrodes should be placed to the back of the head when recording VEPs with EEG. These electrode locations are identified with the letter O as discussed in section 1.3.

When comparing the VEPs elicited by the stimulation of the *central visual field* and the stimulation of the *peripheral vision* by the same stimuli, it can be seen that the stimulation of the central visual field produces larger VEPs [21]. In other words, the stimulus that a subject is looking at elicits larger VEPs than those stimuli that are not in the very centre of the gaze. Therefore, it is possible to present multiple visual stimuli to a subject and detect which stimulus is the subject looking at.

To make the detection of VEPs easier, steady-state visual evoked potentials (SSVEPs) are used. If the visual stimulus is presented at a constant rate and the rate is so fast that the visual pathway does not have enough time to fully recover between stimulus presentations, then the elicited response becomes continuous and it is called SSVEP [48]. See figure 1.5 for an example of VEPs and SSVEPs.

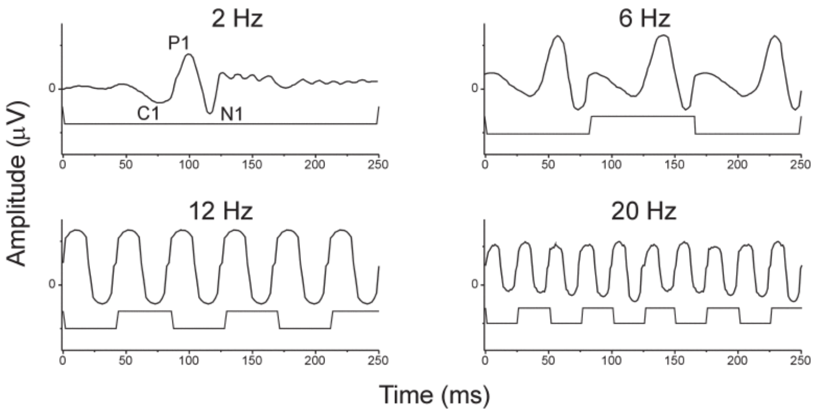


Figure 1.5: VEPs and SSVEPs elicited by stimuli with different frequencies [48, p. 259].

An SSVEP is composed of VEPs that are elicited one after another in a certain rate.

An SSVEP is continuous and therefore it is easier to detect it in the EEG recording than VEPs, which are not continuous. Detecting continuous response is easier and more efficient because it is always present in the recording. Thus usually SSVEPs are elicited instead of VEPs when the goal is to efficiently communicate with external device through brain responses to visual stimuli.

1.5 Choosing neuroimaging device

In section 1.2 different functional neuroimaging methods were compared and it was concluded that currently EEG is most suitable for designing inexpensive and portable interface for controlling a robot. But there is a wide variety of EEG devices available. The aim of this section is to compare some of the devices and choose a user-friendly device that is available to the consumer.

A continuous signal cannot be directly represented in computers or in other digital devices and therefore a continuous signal has to be converted to a *digital signal* to process it with a computer. A digital signal can be acquired from continuous signal by measuring values of continuous signal at a constant rate.

As already mentioned in section 1.3, EEG measures brain activity as a function of voltage versus time. This function represents digital signal, which means that both voltage and time are discrete. In other words, there is a finite set of possible values that the voltage can have and these values are acquired one after another in a certain rate. This rate at which digital values are extracted from a continuous signal is called *sampling rate*. The device which converts continuous voltage to a sequence of discrete values is called analog-to-digital converter (ADC) device. ADC resolution shows how many different values the device can represent.

The table 1.2 shows a comparison of different EEG devices. Devices are compared by price, the number of electrodes or channels, sampling rate and ADC resolution. The higher the sampling rate, the more values are extracted in same time interval. The higher the ADC resolution, the more different voltages the device can represent.

	Price	Channels	Sampling rate	ADC resolution
Mindwave[6]	\$80	1	512 Hz	12 bit
Emotiv EPOC[7]	\$400	14+2	128 Hz	16 bit
OpenBCI[8]	\$450	8	adjustable	24 bit
Mitsar 202[9]	\$10,500 [10]	31+1	2 kHz	24 bit
atiCHamp[11]	\$77,100	160	25 kHZ	24 bit

Table 1.2: Comparison of EEG devices.

From the more consumer-friendly devices, Emotiv EPOC seems to offer good price-quality relationship. Emotiv EPOC has 16 electrodes, two of which are reference electrodes.

[6] http://store.neurosky.com/products/mindwave-1

[7] https://emotiv.com/epoc.php

[8] http://openbci.myshopify.com/products/openbci-8-bit-board-kit

[9] http://www.mitsar-medical.com/eeg-machine/eeg-amplifier-compare/

[10] http://www.novatecheeg.com/products–software.html

[11] http://www.brainvision.com/files/actiCHamp-PyCorder-Flyer_US.pdf

Reference electrodes have two different possible locations: P3 and P4 or locations behind the ears. Other electrodes have fixed locations: AF3, AF4, F3, F4, F7, F8, FC5, FC6, P7, P8, T7, T8, O1, O2. See figure 1.6 for illustration. Emotiv EPOC has a sampling rate of 128 Hz, an internal sampling rate of 2,048 Hz and an ADC resolution of 16 bit. High internal sampling rate is used to remove high frequency artefacts from the signal. The signal is filtered and then transmitted to wireless receiver with reduced sampling rate of 128 Hz.

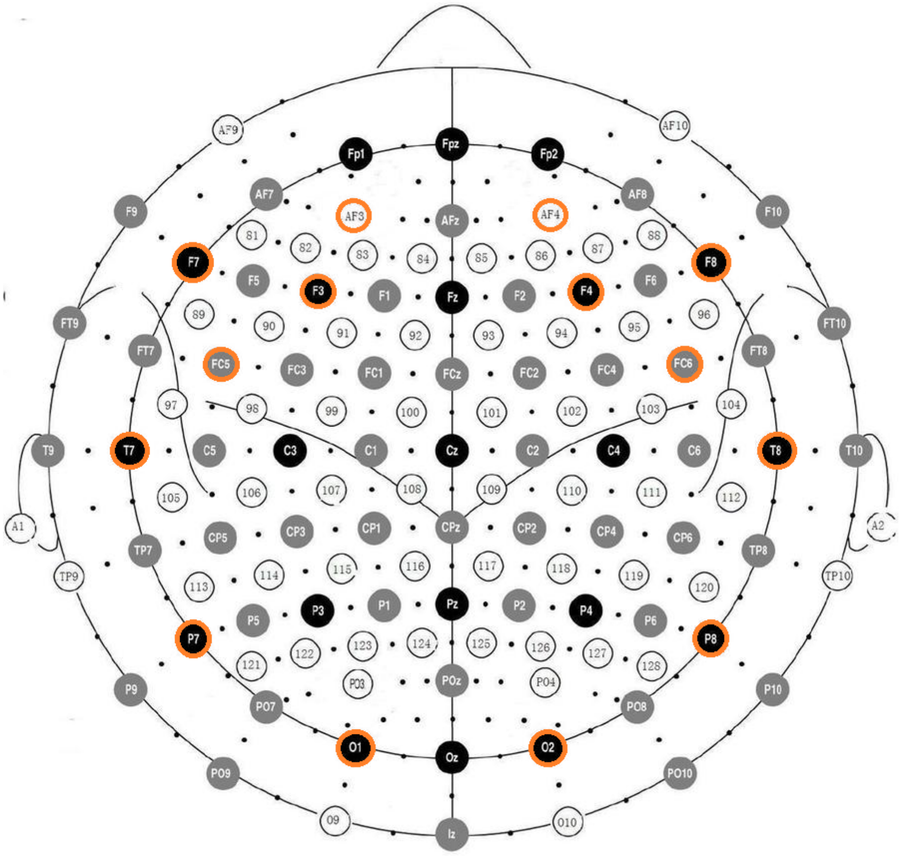


Figure 1.6: Electrode locations used by Emotiv EPOC[12]. Used locations are marked with orange circle.

Research has shown that Emotiv EPOC performs significantly worse than a medical-grade device [14]. But the performance of Emotiv EPOC is good enough to detect SSVEPs in the recording [30, 53, 27, 24]. It has also been shown that it is possible to detect SSVEPs in the Emotiv EPOC recording even when the subject is walking during the recording, despite the artefacts that the movement produces [27]. Walking during the recording session causes the device to move on subject's head and this movement produces artefacts. Thus the performance of Emotiv EPOC is sufficient for detecting SSVEPs and it is chosen as an EEG device for controlling a robot in this thesis.

[12] http://emotiv.wikia.com/wiki/Emotiv_EPOC

2 Implementing VEP-based brain-computer interface

The previous chapter discussed the biology of the brain and described the brain potential called SSVEP. The aim of this chapter is to describe which kind of visual stimuli can be used to elicit an SSVEP and how to detect the elicited SSVEP in the EEG recording. This knowledge is needed to extract information from the EEG recording and use this information to control a robot. In other words, this chapter will discuss how to implement a VEP-based brain-computer interface (BCI).

2.1 Designing visual stimuli

As discussed in section 1.4, it is possible to present multiple visual stimuli to a subject and detect, which stimulus is the subject looking at. It was also mentioned that computer monitor can be used to present visual stimuli. This section will discuss how stimuli with certain blinking frequency can be displayed by a computer monitor.

Research has shown that LCD screens produce more reliable SSVEP response than light-emitting diode (LED) [11]. Another study, however, concludes that BCIs that use LEDs as visual stimuli have achieved better performance [52]. But since using LEDs requires dedicated hardware, only the stimuli that can be displayed by a computer monitor are discussed in this thesis.

A computer monitor has certain size, resolution and *refresh rate*. Monitor resolution and size limit the size of the visual stimuli that can be used and the distance between the stimuli on the screen. Monitor refresh rate, on the other hand, limits the presentation rate or blinking frequency of the stimulus that can be used. Research has shown that LCD screens produce more reliable SSVEP response when using monitor refresh rate for measuring the time between stimuli presentations rather than a timer [11]. Therefore in this thesis monitor refresh rate is used for timing and synchronisation.

Monitor refresh rate is the number of consecutive images or *frames* shown on the screen in a second, assuming that the frames are produces at least as fast as they can be displayed. A frame is one of the images that compose the changing picture on the screen. Monitor with a refresh rate of 60 Hz can display 60 frames per second.

Often blinking one-coloured squares on one-coloured background are used as visual stimuli in SSVEP-based BCIs [52]. The squares may have symbols on them, for example letters or numbers. The stimuli of VEP-based BCI are also called *targets* and the blinking of a target is called *flickering*.

In every frame each target can be in one of two *states*—displayed or not displayed. The state of a target can be switched only when the current frame is replaced with the next frame. The state switches should be distributed as evenly as possible for the target frequency to be constant. Distributing the state switches is easier with some frequencies than others. For example, if refresh rate is 60 Hz, then the state switches for 10 Hz, 11 Hz and 12 Hz target should be distributed as follows:

- 10 Hz flickering can be achieved by presenting the target $\frac{60}{10} = 6$ times slower than the refresh rate. This means, that the target has to be presented once in every 6 frames. Since 6 is an even number, 10 Hz flickering can be achieved by changing the state of the target after every $\frac{6}{2} = 3$ frames. If this flickering is plotted as a function of state versus time as in figure 2.1, it can be seen that it produces a *square wave*. In this thesis, the waveform produced by plotting the flickering is called *flickering waveform*.

- 12 Hz flickering can be achieved by presenting the target $\frac{60}{12} = 5$ times slower than the refresh rate. The target has to be presented once in every 5 frames. Since 5 is an odd number, the amount of time the target is in displayed state and in not displayed state cannot be equal. Therefore, the target should be 3 frames in displayed state and 2 frames in not displayed state or the other way round. If representing this flickering as a flickering waveform as in figure 2.1, it can be seen that it produces a *rectangular wave*.

- 11 Hz flickering can be achieved by presenting the target $\frac{60}{11} \approx 5.45$ times slower than the refresh rate. This means, that the target has to be presented once in every 5.45 frames. Since 5.45 is not a natural number, the flickering will be irregular. 11 Hz target flickering from the paper by Wang *et al.* [43] is used as an example in figure 2.1. Although 11 Hz frequency produces irregular flickering, it is still possible to detect SSVEP elicited by it in the Emotiv EPOC recording [30].

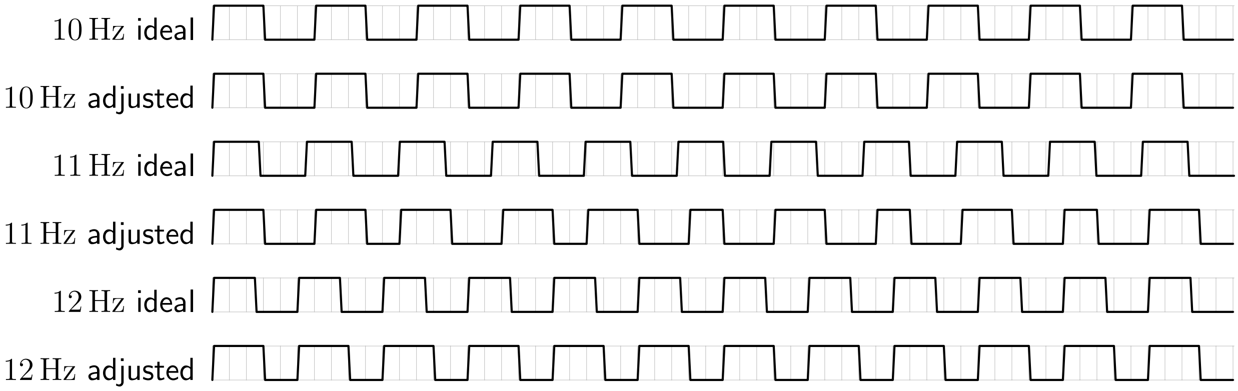
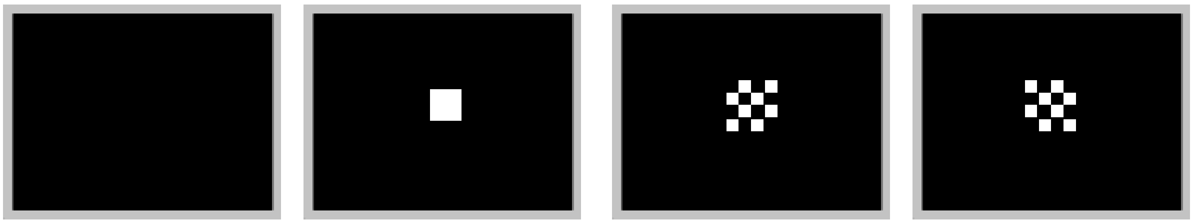


Figure 2.1: Adjusting target flickering to 60 Hz refresh rate. The black line represents the flickering as state versus time. The states are displayed and not displayed. Vertical grid lines represent frame changes. There are 60 vertical grid lines and thus this figure shows the state changes of targets with different frequencies in one second.

A *duty cycle* is used to characterise a rectangular wave. Duty cycle is the percentage of the amount of time the target is in displayed state in one period. If the target is in displayed state for 2 frames and in not displayed state for 3 frames in one period, then the duty cycle of the rectangular wave is $\frac{2}{2+3} \cdot 100\% = 40\%$. Square wave has a duty cycle of 50%. Research has shown that the SSVEP elicited by a square wave flickering can be more accurately detected than those elicited by a rectangular wave flickering [52].

The previous discussion was about blinking shapes or *single graphic* stimuli. There is another type of SSVEP stimuli called *pattern reversal* stimuli that can also be displayed on a computer screen. Pattern reversal stimuli is rendered by changing between two different patterns, for example alternating the colours of a chequerboard [52]. See figure 2.2 for an example of single graphic and pattern reversal stimuli.



(a) The states of a single graphic stimulus. (b) The states of a pattern reversal stimulus.

Figure 2.2: Different types of visual stimuli. A stimulus alternates between the two given states.

The main difference between single graphic stimuli and pattern reversal stimuli is that single graphic stimuli elicits SSVEP response after every two alterations, while pattern reversal stimuli elicits SSVEP response after every alteration [52]. The fastest possible target frequency can be achieved by changing between the states of a target every time a new frame is displayed. If the state is changed at lower rate, the target frequency will also be lower. Target frequency can be calculated as follows:

$$f_{single\ graphic} = \frac{n}{2T} \qquad f_{pattern\ reversal} = \frac{n}{T} \tag{2.1}$$

where T is the period of the flickering waveform, n is the number of times the target state is switched in a time period of T and f is the flickering frequency.

A computer monitor can be used to present visual stimuli to a subject and the refresh rate of the monitor should be used for measuring the time between stimuli presentations. Adjusting target flickering to the rate at which the frames are changed can produce different flickering waveforms for different flickering frequencies. Since the flickering waveforms are different, the SSVEP responses are also different.

2.2 Overview of Fourier analysis

As discussed in the previous section, the flickering of a target can produce different waveforms. This section will discuss how these waveforms and other signals can be represented by sums of simpler trigonometric functions. The study of this decomposition process is called Fourier analysis, named after Joseph Fourier, whose insight to model all functions by trigonometric series was a breakthrough in the field in 1807.

The following paragraph is based on the book by Hartmann [18]. The simpler trigonometric functions that a signal is decomposed into are *pure tones*—the waveforms that contain only one frequency. All other waveforms contain at least two frequencies. Pure tone waveforms are sine and cosine waves. Important property of a pure tone is that linear operations do not change the shape of the pure tone waveform. See figure 2.3 for an example of a pure tone.

The pure tones are used to represent all possible frequencies that a signal may contain. The decomposition process of a signal is called *Fourier transform* and it is used to decompose a function of time into pure tones or *frequency components* that make it up. The function of time can be for example the EEG recording represented as voltage versus time or the target flickering represented as state versus time. Fourier transform converts signal from time domain or the function of time to frequency domain or the function of frequency. The representation of a time-domain signal in a frequency domain is called *frequency spectrum*. Frequency spectrum contains information about amplitude and phase of different frequency components. Therefore, frequency spectrum can be presented as a function of frequency versus amplitude and phase.

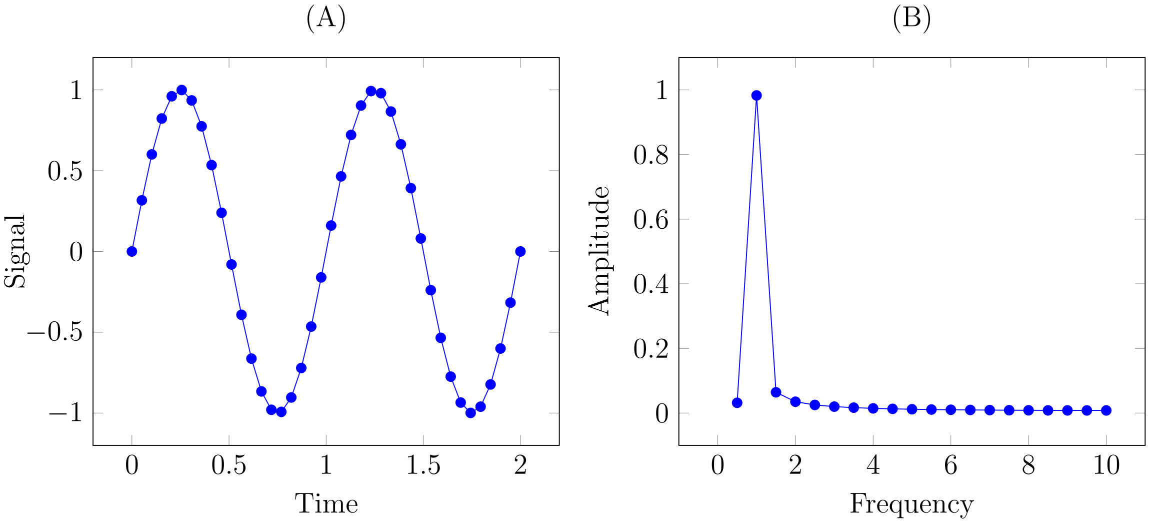


Figure 2.3: A sine wave in time domain (A) and its frequency components' amplitudes (B). Theoretically pure tone has only one frequency, but since the signal in time domain is digital, it is an approximation of the actual pure tone and therefore the representation of frequency components' amplitudes is not perfect.

To represent both amplitude and phase, complex numbers are used. The amplitude and phase do not correspond to the real and imaginary part of the complex number but rather are related to the absolute value or the modulus and the phase of the complex number. In this thesis the phase information from the frequency spectrum is not used. There are, however, BCIs that also use the phase information [40]. Since only the information about amplitude is required in this thesis, it is possible to convert the frequency spectrum into *power spectral density*, which can be represented as a function of amplitude squared versus frequency.

To conclude previous discussion, the amount of frequency f present in a signal can be calculated by first calculating the frequency spectrum with Fourier transform and then taking modulus squared of the frequency spectrum's value at frequency f.

In digital devices, however, theoretical power spectral density cannot be calculated. The measurement period would have to be infinitely long to a acquire the true power spectral density [8]. Therefore, spectral estimation is used. The modulus squared of frequency spectrum in a real-world application is called *periodogram* and it is the estimation of the power spectral density. There are other spectral estimation methods available.

Since digital devices work with digital signals as discussed in section 1.5, in a real-world application discrete version of the Fourier transform is used. The algorithm used to compute the discrete Fourier transform is called fast Fourier transform (FFT). The frequency spectrum calculated by FFT is discrete—if the digital real-valued input signal, as is the case with EEG recording, has N values then the output has the integer part of $\frac{N}{2}$ values. This derives from the definition and symmetric property of the discrete Fourier transform.

The signal that is recorded, however, may contain frequencies that are too high to be detected from the digital signal. Theoretically, the sampling rate of the device has to be more than two times higher than the highest frequency in the signal to reconstruct the continuous signal from the recorded digital signal and also to decompose the signal into frequency components. The highest frequency that can be detected with a sampling rate of f is $\frac{2}{f}$ and it is called *Nyguist frequency*. But since real-world application are imperfect, even higher sampling rate is needed. For example, as discussed in section 1.5, Emotiv EPOC has internal sampling rate of 2,048 Hz to more accurately record frequencies up to its Nyguist frequency of $\frac{128\,\text{Hz}}{2} = 64\,\text{Hz}$, since the actual sampling rate is 128 Hz.

To conclude previous discussion, the frequency spectrum calculated by FFT from real-valued time-domain signal with N values is defined at frequencies $\frac{1}{N}, \frac{2}{N}, \ldots \frac{f/2}{N}$. These frequencies are also called *frequency bins*. The length of a frequency spectrum depends on the length of the signal from which it is calculated. The longer the input signal, the more frequency bins will be acquired.

Thus a time-domain signal can be decomposed into pure tones or sine and cosine waves using FFT. FFT calculates the frequency spectrum of a time-domain signal. Frequency spectrum can be converted to the estimation of power spectral density which contains only the information about the amplitudes of the pure tones. Analysing the power spectral density can be used to detect SSVEPs in an EEG recording.

2.3 Decomposing target flickering

This section will focus on the decomposition of the flickering waveforms, which were described in section 2.1. The decomposition process was discussed in section 2.2. As discussed in previous section, only sine and cosine waves are composed of a single frequency. Other waveforms have more frequency components.

The frequency component of a signal that has the lowest frequency among all the frequency components of the signal is called *fundamental* or the first *harmonic* of the signal. Harmonic of a signal is a frequency component with frequency that is an integer multiple of the fundamental frequency of the signal. If the fundamental frequency is f, then first, second are third harmonics have frequencies of $1f$, $2f$, $3f$ respectively.

It can be shown that a square wave is composed of its odd harmonics. If the flickering waveform is a square wave or a rectangular wave, the fundamental frequency of the waveform is the frequency of the stimuli presentation. Therefore, a square wave with fundamental frequency f can be represented as a sum of sine waves

$$\text{square}(t) = \sum_{n=1,3,5,\ldots}^{\infty} \frac{1}{n} \sin(2\pi n f t) \tag{2.2}$$

where square(t) is the flickering represented as state versus time, $\frac{1}{n}$ is the amplitude of a frequency component and f is the fundamental frequency of the waveform. See figure 2.4 for an example of a square wave, three of its harmonics and their amplitude spectrum. It can be seen that the fundamental of the square wave has the highest amplitude among its frequency components. Rectangular wave's frequency components depend on the duty cycle of the wave. But as is the case with square waves, the fundamental has the highest amplitude also in a rectangular wave.

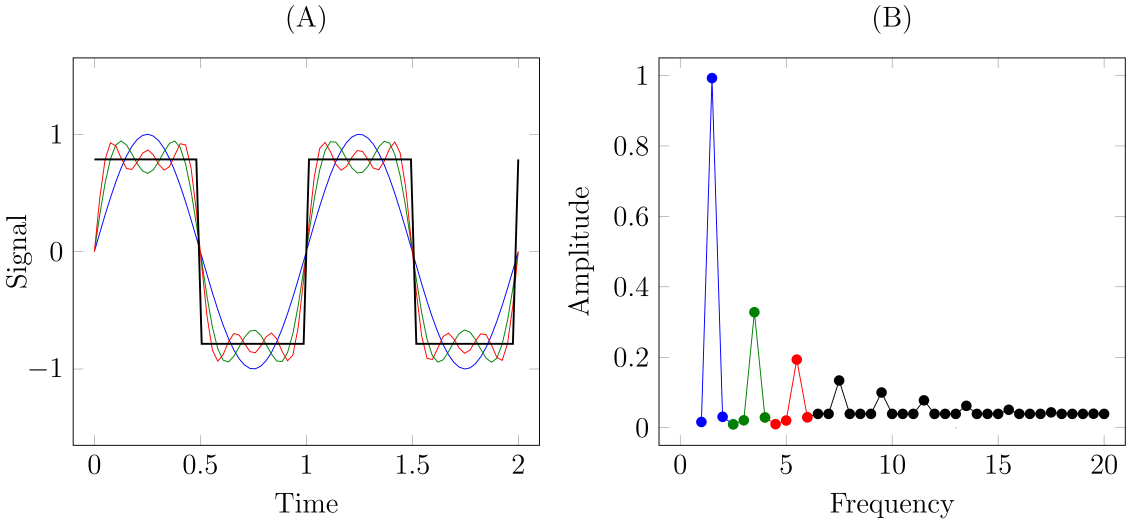


Figure 2.4: A square wave in time domain (A), its first harmonic (blue), the sum of its first two harmonics (green) and the sum of its first three harmonics (red). The amplitude spectrum (B) shows the amplitudes of the same signals with the same colour-coding. Please note that the green signal also contains the blue signal, the red signal contains both blue and green signal and black signal contains all the previous signals. In general, every next signal in amplitude spectrum also contains the previous signals.

Unfortunately, the SSVEP response to target flickering does not contain only the frequencies that are present in the frequency components of the flickering waveform—SSVEP contains other frequencies too. However, the frequencies that are present in the frequency components of the flickering waveform are more successfully elicited in SSVEP response [39].

Generally an SSVEP contains frequency components with frequencies that are integer multiples of the flickering frequency [20]. Therefore, target frequencies should be chosen so that neither of these frequencies is an integer multiple of the other. Otherwise it might not be distinguishable which target flickering elicits which frequency in the SSVEP response. For example, if 6 Hz and 12 Hz Hz flickering frequencies are both used, then 6 Hz flickering also elicits 12 Hz frequency in the SSVEP since 12 Hz is the second harmonic of 6 Hz flickering and thus it might not be possible to distinguish this 12 Hz component from the 12 Hz component elicited by 12 Hz flickering.

It has even been reported that SSVEP contains frequency components that have lower frequency than the fundamental of the flickering waveform [20]. These frequency components have frequency of $\frac{f}{n}$, where f is the fundamental frequency of the flickering waveform and n is a natural number. But since these components have very small amplitudes, these components are not used in SSVEP-based BCIs.

An SSVEP reflects certain properties of the visual stimulus. The frequencies that are present in the flickering waveform are likely to be found also in the SSVEP response, but other frequency components are present in the SSVEP too [39]. It is sufficient to detect only the frequency component with the same frequency as the target flickering in a SSVEP-based BCI, but to improve the performance of the BCI other frequency components should be detected too [31].

2.4 Evaluating the performance of a brain-computer interface

The most commonly used method for evaluating the performance of a BCI is called information transfer rate (ITR) [47]. ITR was defined by Wolpaw *et al.* [44] in 1998

$$B = \log_2 N + P \log_2 P + (1 - P) \log_2[(1 - P)/(N - 1)] \tag{2.3}$$

where B is the ITR in units of bits per trial or bits per command, N is the number of targets and P is the accuracy or the probability that the user's choice is actually selected. To make the units of ITR more understandable, ITR is calculated in units of bits per minute [44]

$$B_t = B \cdot (\frac{60}{T}) \tag{2.4}$$

where B_t is the ITR, B is calculated with equation 2.3 and T is the time needed to identify a chosen command.

There are, however, preconditions that have to be fulfilled in order to calculate correct ITR with the previously given equations. These preconditions are the following: BCI is memoryless, all commands are equally likely to be chosen, the accuracy of choosing a target is the same for every target and in case of choosing a wrong target, all wrong targets are equally likely to be chosen [47].

If these preconditions are met, the ITR of a BCI can be calculated using equation 2.4. Using the same evaluation method allows the performance of different BCIs to be easily compared.

2.5 Improving the accuracy of a spectrum with signal processing

The aim of this section is to describe some digital signal processing techniques that can improve the performance of a SSVEP-based BCI by improving the accuracy of an amplitude spectrum or power spectral density.

2.5.1 Detrending the signal

A linear *trend* or steady increase or decrease of values in an EEG recording can make the detection of SSVEPs less accurate. As discussed in section 2.2, Fourier transform is used to decompose a signal into frequency components. If the signal has a trend, the trend will also be decomposed. The decomposed trend will not provide any useful information and it makes detecting the actual SSVEP less accurate. A comparison of amplitudes acquired from a signal with trend and from the same signal without trend can be seen in figure 2.5.

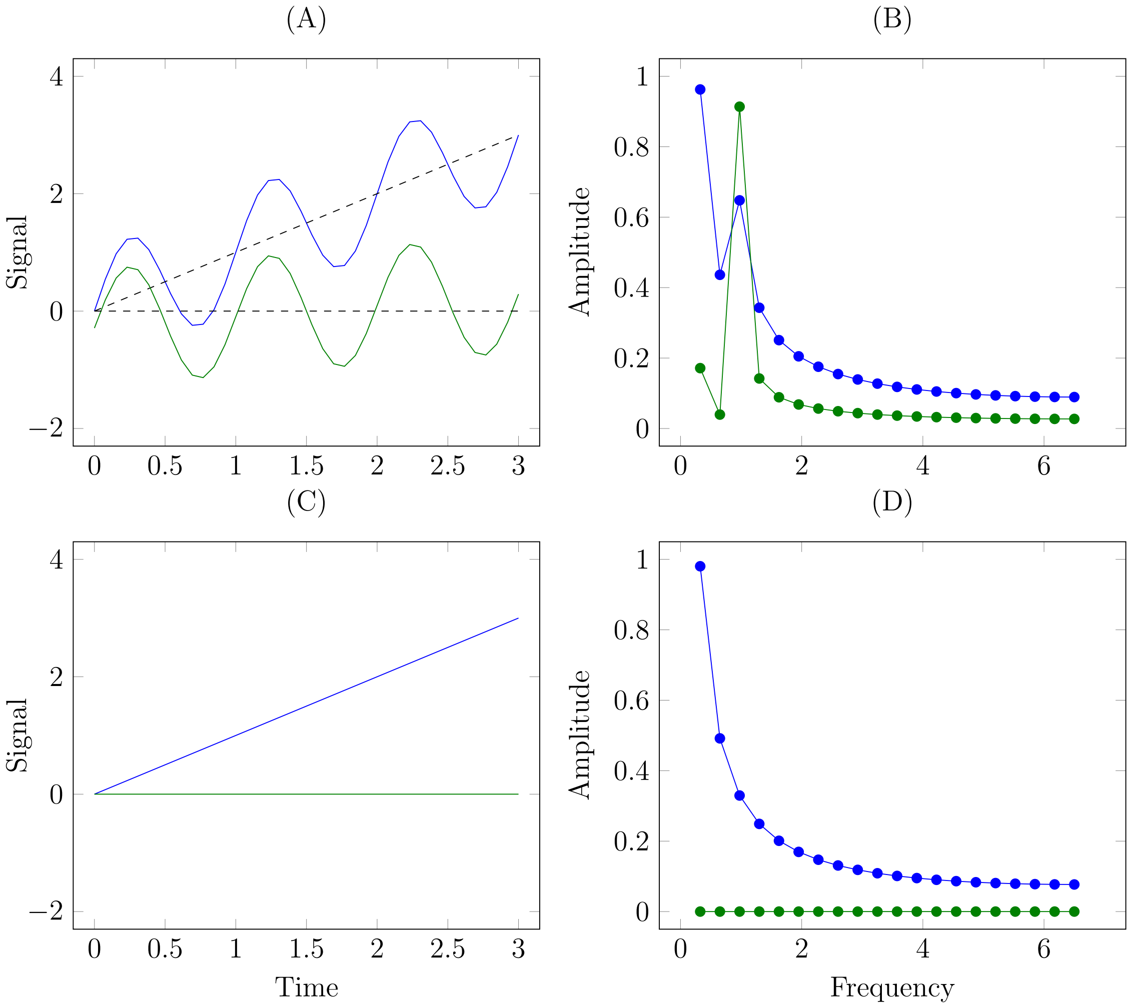


Figure 2.5: A signal with trend (blue) and the same signal without trend (green) in time domain (A). The amplitude spectrum (B) shows the amplitudes of the frequency components of the same signals with the same colour-coding. The plots in the second row show the trends of the signals in time domain (C) and the amplitude spectrums of these trends (D). The green signal in time domain does not have a trend so its trend is presented as a constantly zero function.

Removing trend from a signal is called *detrending*. The average value or the *mean* of a detrended signal is zero. The detrending also works if there is no steady increase or decrease of values in the signal. In this case, just a constant value—the mean of the signal—is subtracted from all the values of the digital signal and as a result, the mean of the signal will be zero. Subtracting a constant value from the signal does not change the amplitudes of the frequency components of the signal and it does not add additional frequency components.

A linear trend can also be removed from the signal in segments—this means that the signal is divided into equal length segments and trend is removed from every segment separately. This is useful if the trend changes in the signal. Detrending a signal does not decrease but can increase the accuracy of detecting SSVEPs. Therefore the EEG recording should be detrended before performing FFT.

2.5.2 Windowing the signal

When estimating the power spectral density of a signal using FFT, it has to be decided how many values will be recorded before performing FFT on the acquired samples or in other words, how long *window* will be used. This means that the signal is divided into segments or in a sense the signal will be looked at through a window. Window function is a function that has non-zero values in a certain range and its value is zero outside that range. Windowing means that a signal is multiplied with a window function. The multiplication is element-wise

$$(w \cdot s)(x) = w(x) \cdot s(x) \tag{2.5}$$

where s is signal segment ans w is window function.

The non-zero values of a window function usually increase until the centre of the range, at the centre there is the highest value and then the values start decreasing again until the end of the range. See figure 2.6 for an example of hanning window. The general purpose of multiplying a signal with a window function is to smooth the start and the end of the signal. If the recorded signal has a clear periodic component but the signal is divided into segments so that one segment does not contain exactly integer multiple of periods of the component, then phenomena called *spectral leakage* will happen. Spectral leakage means that some of the power of the periodic component will be distributed over other frequency bins and thus the correct frequency bin will have less power and incorrect frequency bins will have more power. Smoothing the start and the end of a signal is used to minimise the spectral leakage.

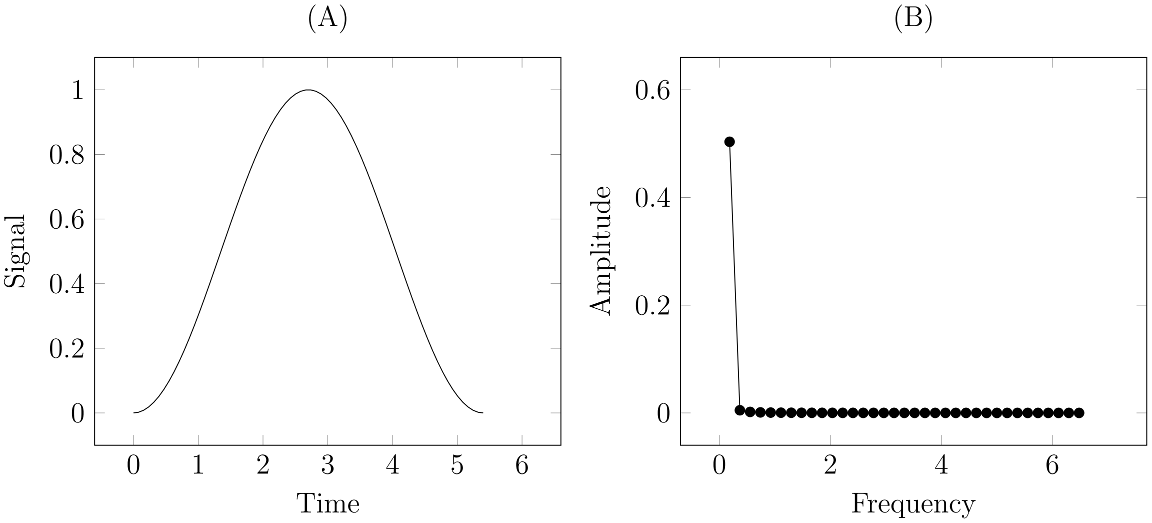


Figure 2.6: Hanning window in time domain (A) and its amplitude spectrum (B).

Multiplying a signal with a window function requires the signal to have zero mean. Otherwise some unwanted components will appear in the estimated power spectral density.

If multiplying the signal $s(x)$ that can be presented as a sum of detrended signal $z(x)$ and the trend $t(x)$ with window $w(x)$, then

$$(w \cdot s)(x) = w(x) \cdot (z(x) + t(x)) = w(x) \cdot z(x) + w(x) \cdot t(x) \tag{2.6}$$

It can be seen that if the trend $t(x)$ is not constantly zero, then windowing adds a component $w(x) \cdot t(x)$ to the signal in addition to changing the amplitudes of the components of $z(x)$. If $t(x)$ is constantly zero, then the signal has zero mean and no additional component is added. Therefore, signals with non-zero mean or trend should be detrended before windowing. Another thing to keep in mind is that windowing a signal makes the peaks in the power spectral density wider. See figure 2.7 for an example of sine waves with zero mean and non-zero mean windowed with hanning window and a comparison of amplitudes acquired from actual signal and windowed signal.

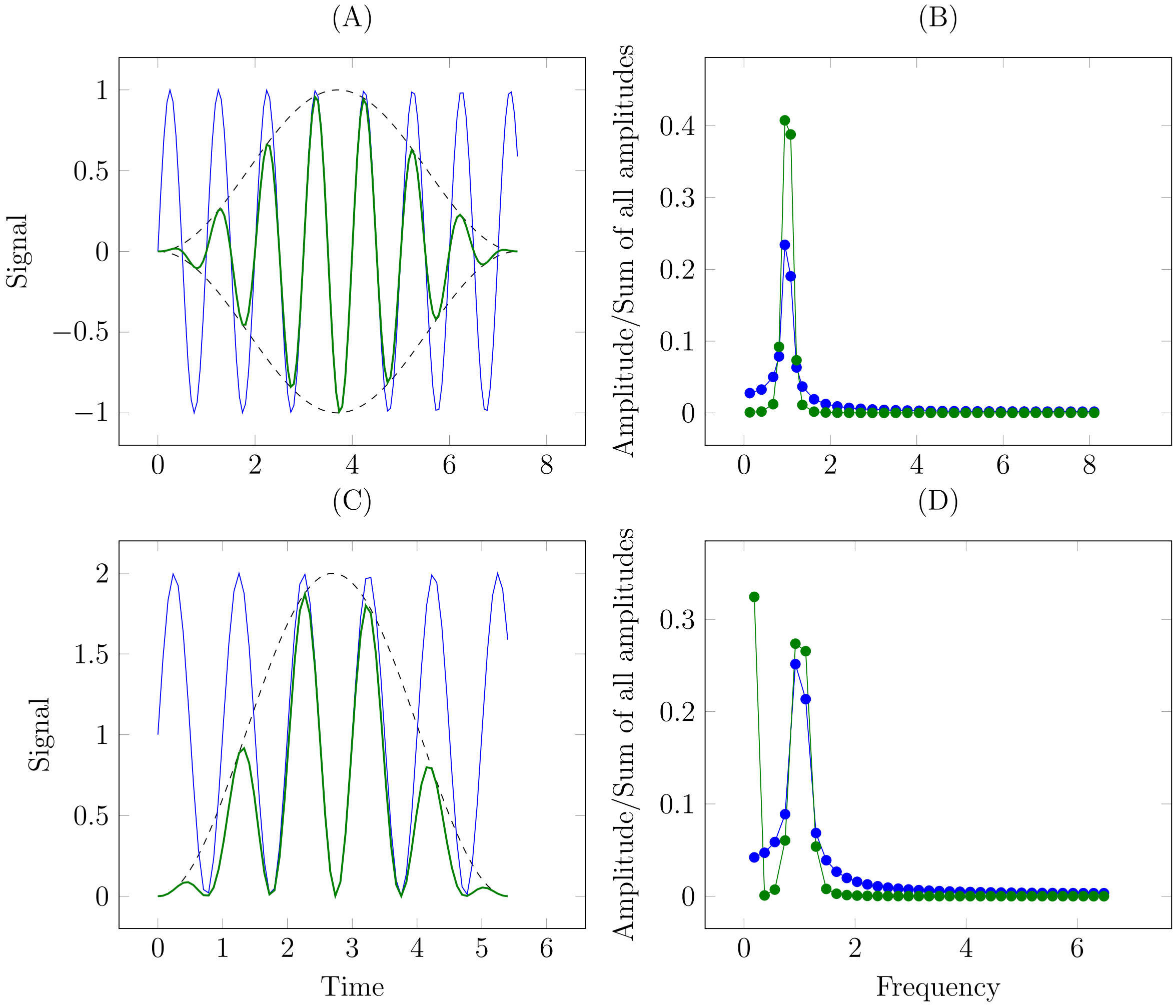


Figure 2.7: A signal with zero mean (blue) and the same signal multiplied with window function (green) in time domain (A). The amplitude spectrum (B) shows the amplitudes of the frequency components of the same signals with the same colour-coding. Plots in the second row show the same information for a signal with non-zero mean.

The longer the window, the more values will be acquired and the estimated power spectral density will be closer to the actual power spectral density [8]. This was already briefly mentioned in section 2.2. However, to control a robot it is necessary for the BCI to detect SSVEPs as fast as possible and therefore the window length should be as short as possible. Thus, choosing the right window length is important in designing a fast and accurate BCI.

2.5.3 Zero padding the signal and approximating unknown values

Another problem that occurs when analysing power spectral density is that the power spectral density of a digital signal is discrete and therefore it might not have frequency bins at the exact values of interest. For example, if designing a BCI, the frequencies of interest are the targets frequencies or integer multiples of the targets frequencies, because these are the frequency components of SSVEP as discussed in section 2.3.

Interpolation can be used to approximate the value between frequency bins to calculate the power of a frequency that does not correspond to any frequency bin. In general, interpolation is used to construct a digital signal between its discrete values or in other words to approximate the continuous signal from which the discrete values were extracted from. Therefore, interpolation can also be used to approximate the EEG recording if some of the values are lost in the process of sending values from the recording device to the computer.

In the paper by Hakvoort *et al.* [17], linear interpolation was used to approximate the amplitude of frequencies that did not correspond to any frequency bin. However, linear interpolation is not the best option to estimate peaks in power spectral density. In figure 2.8 there is a comparison of amplitude estimation of linear interpolation and barycentric interpolation.

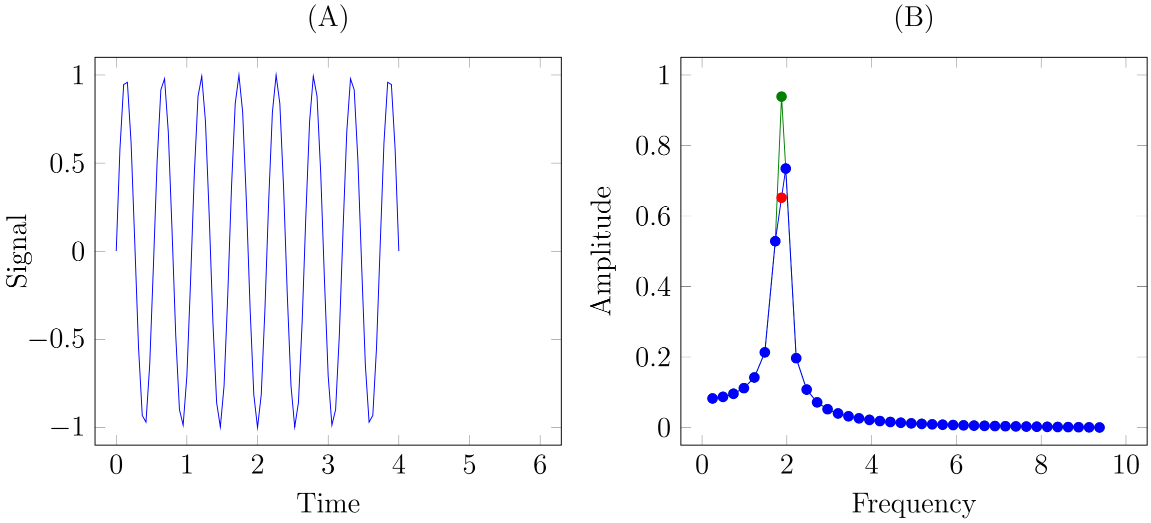


Figure 2.8: A signal in time domain (A) and its amplitude spectrum (B). The amplitude spectrum shows the comparison of the amplitude approximation of 1.85 Hz frequency component using linear interpolation (red) and barycentric interpolation (green).

Another possibility to solve the lack of frequency bins problem is to use *zero padding*. Zero padding means that zeros are added to the end of the signal before performing FFT. This results in more frequency bins in the power spectral density, because the number of frequency bins depends on the length of the input signal as discussed in section 2.2. The only alteration in power spectral density is that it has more frequency bins if calculated from a zero-padded signal. The comparison of amplitudes acquired from a signal and from the same signal with zero padding can be seen in figure 2.9. Multiplying a signal with a window function before zero padding results in a smoother transition between the signal and its zero padding.

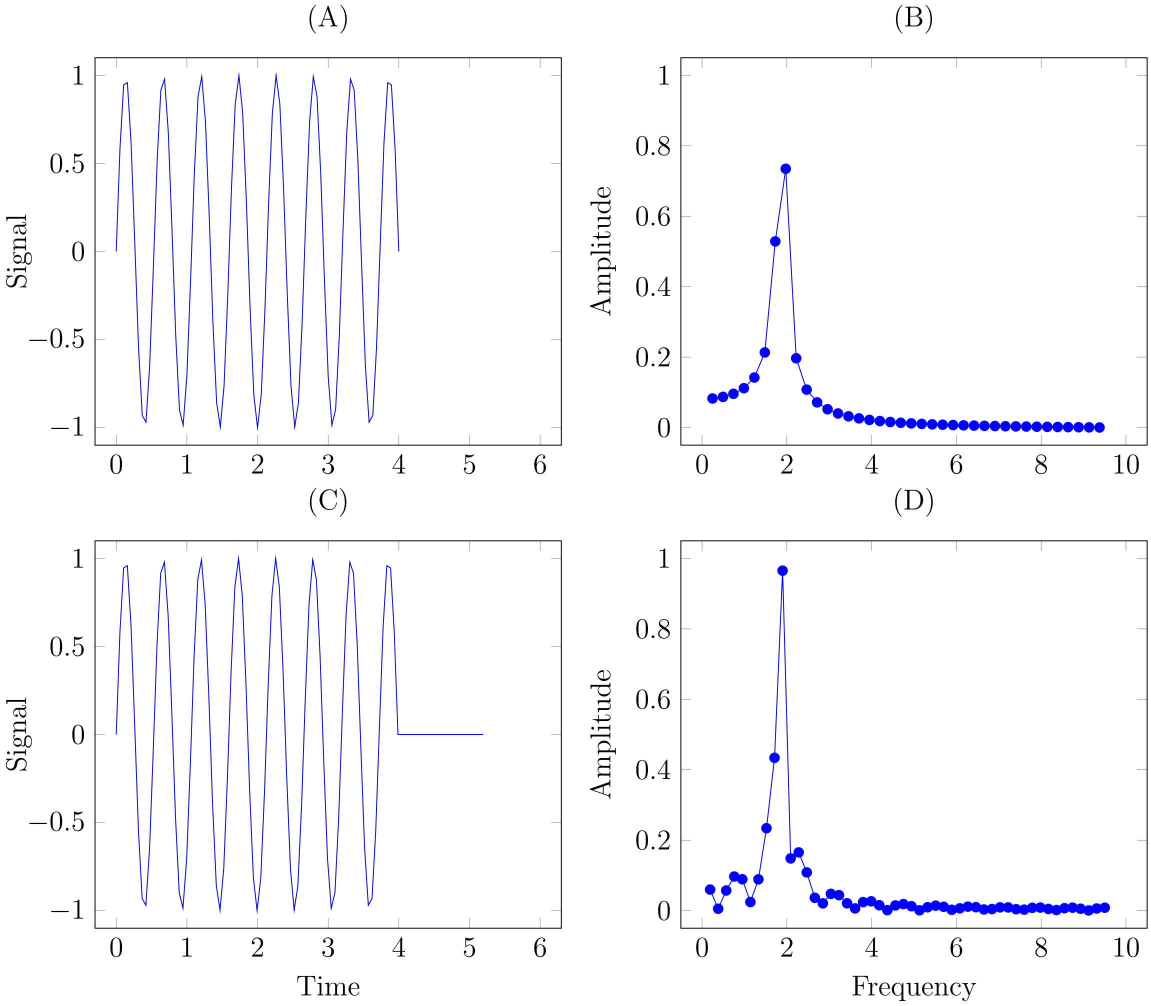


Figure 2.9: A signal in time domain (A) and the same signal zero padded (C). Plots in the second column show the amplitude spectra of the corresponding signals in the first column.

To sum up, the signal should be windowed or in other words multiplied with a window before zero padding. Windowing a signal requires the signal to have zero mean as discussed in the previous section and therefore the signal should be detrended before windowing. Thus the correct order of using signal processing techniques described in this chapter is: interpolating to approximate lost packets or digital signal values, detrending, windowing, zero padding and then interpolating to approximate values between frequency bins.

2.6 SSVEP-based BCI feature extraction methods

The aim of this section is to describe two methods used for detecting SSVEPs in an EEG recording. These methods are called *feature extraction* methods. This section describes the PSDA and CCA feature extraction methods. These are the methods used in chapter 3 to design an application for controlling a robot.

2.6.1 Power spectral density analysis method

This section describes SSVEP-based BCI feature extraction method called PSDA. PSDA is widely used in SSVEP-based BCIs [5]. This method is based on a power spectral density estimation; one of the estimation methods called periodogram was discussed in section 2.2.

There are different ways how to use the power spectral density estimation for feature extraction. Some BCIs use peak finding to find highest values in the power spectral density [28]. Other BCIs use a training session to find a threshold value that certain frequency's amplitude or power has to exceed in order to select the target with corresponding frequency as user's choice. This method requires a training session during which the threshold values are determined, but SSVEP-based BCIs could be implemented without the need for a training session. Threshold values, however, can be used together with other methods.

BCIs that use PSDA feature extraction usually calculate signal-to-noise ratio (SNR) or other values that are related to SNR of each target and then select the target with the highest SNR or related value as user's choice. For example, the ratio of frequency's power to the mean of adjacent frequency bins can be calculated to select user's choice [4]

$$\mathrm{SNR}(f_k) = \frac{2P(f_k)}{P(f_{k+1}) + P(f_{k-1})} \tag{2.7}$$

where P is the function representing power spectral density and $f_1, \dots, f_k, \dots, f_m$ are the frequency bins of the periodogram in increasing order. The SNR as defined in equation 2.7 can be calculated for every target frequency and the target that has the frequency with highest SNR is assumed to be the user's choice. This works because SSVEP has a component with the same frequency as the target frequency as discussed in section 2.3.

More than two adjacent frequency bins can be used to calculate the SNR [46]. Therefore, the equation 2.7 can be generalised

$$\mathrm{SNR}(f_k) = \frac{nP(f_k)}{\sum_{i=1}^{n/2} \big(P(f_{k-i}) + P(f_{k+i})\big)} \tag{2.8}$$

where n is the number of adjacent frequency bins used. To generalise the equation even more, the ratio of the frequency's power to the whole power spectral density or all the frequency bins can be calculated

$$\mathrm{SNR}(f_k) = \frac{P(f_k)}{\sum_i P(f_i)} \tag{2.9}$$

It is also possible to just compare the powers of different target frequencies and select the target with highest power [17]. This is equivalent to using equation 2.9.

All these methods can use in addition to the target frequency also its integer multiples to improve the performance of the BCI, as mentioned in section 2.3. In this case the sum of the SNRs can be used

$$\sum_{i=1}^{h} \mathrm{SNR}(if_k) \tag{2.10}$$

where h is the number of integer multiples used. Often three integer multiples are used. After calculating the SNR for all the target frequencies and for their integer multiples if necessary, the target with highest SNR or the highest sum of SNRs can be selected as user's choice.

To design a SSVEP-based BCI that uses PSDA feature extraction method it is enough to use one of the methods described in this section. Often SNRs or just the powers of different targets are compared to determine the user's choice.

2.6.2 Canonical correlation analysis method

CCA was first introduced by Harold Hotelling in 1936 [22]. In 2001, CCA was used to introduce a novel method for detecting neural activity in fMRI data [15]. Likewise, CCA was introduced to EEG recording analysis for the first time in 2007 by Lin *et al.* [28]. This section gives necessary background information and describes the method proposed by Lin *et al.*

Overview of basic statistics

Mathematically EEG recordings can be modelled using random variables. This interpretation is necessary to make mathematical statements about the calculated *statistics*, such as the mean and *covariance*. One specific EEG recording can be viewed as a data *sample* or a set of data collected from the continuous signal. Therefore, EEG recording can be represented as a sequence of recorded values $\mathbf{x} = (x_1, x_2, x_3, \dots, x_n)$.

As was the case with power spectral density, the statistics calculated from a data sample are not the exact parameters for the whole EEG signal. Statistics are used to estimate the theoretical values. The sample mean of a data set, for example of an EEG recording can be calculated with

$$m(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} x_i \tag{2.11}$$

To measure the similarity of two EEG recordings, sample covariance can be used. Sample covariance measures how much two data sets change similarly. For two EEG recordings $\mathbf{x} = (x_1, \dots, x_n)$ and $\mathbf{y} = (y_1, \dots, y_n)$ the sample covariance is

$$q(\mathbf{x}, \mathbf{y}) = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - m(\mathbf{x}))(y_i - m(\mathbf{y})) \tag{2.12}$$

The addend $(x_i - m(\mathbf{x}))(y_i - m(\mathbf{y}))$ in the equation 2.12 is positive, if both signals change similarly at the corresponding time point or in other words, if both signals are above or below their means at the same time. The addend is negative, if the signals change in the opposite way or in other words, if one signal is above its mean and the other is below its mean.

Sample covariance can be normalised so that the result will be between 1 and -1. Normalised covariance is called sample *correlation*. Sample correlation can be calculated with

$$r(\mathbf{x},\mathbf{y}) = \frac{q(\mathbf{x},\mathbf{y})}{\sqrt{q(\mathbf{x},\mathbf{x})q(\mathbf{y},\mathbf{y})}} \tag{2.13}$$

There is somewhat different correlation defined for time-domain signals, called *cross-correlation*. Cross-correlation takes also into account the possible shift in time between the signals. For example, the correlation of sine and cosine calculated with equation 2.13 is 0, which means that according to equation 2.13 sine and cosine are uncorrelated despite the fact that sine and cosine waves are actually very similar. Cross-correlation is the function of the measure of similarity versus time lag between the signals

$$(\mathbf{x} \star \mathbf{y})(t) = \sum_i x_i y_{i+t} \tag{2.14}$$

The equations 2.13 and 2.14 can be used to measure the similarity between two EEG recordings. Measuring the similarity between two signals can be used as feature extraction method called *template matching*. The template matching method requires a training session to acquire templates that can be later compared to EEG recording when a subject is using the BCI. Each template shows the state of the brain when the subject is watching certain target. If the template acquired during training session and the current recording are similar enough, the target corresponding to the matching template will be selected.

Designing reference signals

CCA method does not use templates as discussed in the previous section. CCA method uses sine and cosine waves as *reference signals* instead of templates in SSVEP-based BCIs. CCA is a statistical method used to measure the similarity between two sets of signals or two sets of data samples. Therefore, CCA can be used to measure the similarity between multichannel EEG recording and multiple reference signals. In contrast, the equation 2.13 can be used to measure the similarity between only two signals or two data samples.

In SSVEP-based BCIs one set of data is the multichannel EEG recording. For example, if recording data with electrodes located in O1 and O2, the recorded data can be represented as a vector of vectors $\mathbf{X} = (\mathbf{x}_{O1}, \mathbf{x}_{O2})$. The second set of data contains pure tones, each of which has a frequency of integer multiple of a target frequency. There is different set of reference signals for every target, because targets have different frequencies. Both sine and cosine waves are used in the second data set

$$\mathbf{Y} = \begin{pmatrix} y_{sin1}(t) \\ y_{cos1}(t) \\ y_{sin2}(t) \\ y_{cos2}(t) \\ y_{sin3}(t) \\ y_{cos3}(t) \end{pmatrix} = \begin{pmatrix} \sin(2\pi 1 f t) \\ \cos(2\pi 1 f t) \\ \sin(2\pi 2 f t) \\ \cos(2\pi 2 f t) \\ \sin(2\pi 3 f t) \\ \cos(2\pi 3 f t) \end{pmatrix} \tag{2.15}$$

In the method proposed by Lin *et al.* [28] three integer multiples are used as in equation 2.15. The reason why both sine and cosine waves are used is that the phases of the

SSVEP components are not known. As already mentioned in the previous section, sine and cosine waves are uncorrelated according to equation 2.13. Using both sine and cosine waves as reference signals gives optimal minimum correlation of $\frac{1}{\sqrt{2}}$ between a reference signal and an ideal SSVEP component.

The optimal minimum correlation means that if SSVEP component with the same frequency and amplitude as a reference signal but with different phase is compared to both sine and cosine reference, then the maximum of the absolute values of these two correlations will be no less than $\frac{1}{\sqrt{2}} \approx 0.707$. This can be seen when calculating the cross-correlation of the SSVEP component with both signals and taking the absolute value of the resulting values. This is illustrated in figure 2.10.

It is possible to take the absolute value of the correlations because CCA treats correlation and anticorrelation or positive and negative correlation similarly. The cross-correlations will be similar to the one presented in figure 2.10 if the signal has different frequency than the reference signals, but in this case the cross-correlations will have smaller amplitude.

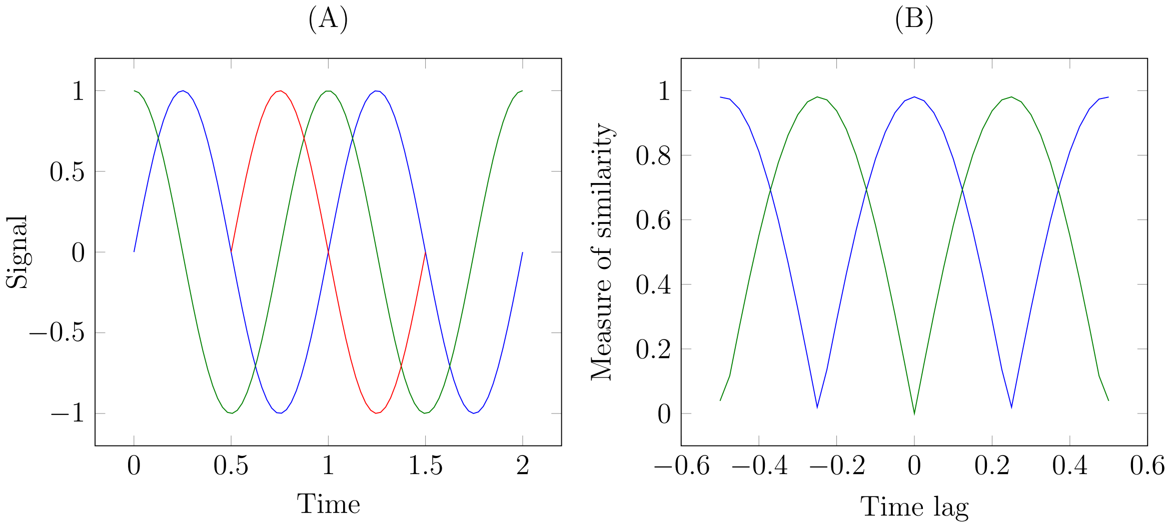


Figure 2.10: A signal (red) and sine (blue) and cosine (green) reference signals in time domain (A). The second plot (B) shows the absolute value of the cross-correlation of the signal with sine wave (blue) and cosine wave (green).

Thus if there is SSVEP component that corresponds to target frequency or its integer multiple, then there is positive correlation between the SSVEP component and at least one of the reference signals.

Comparing two sets of signals

The covariance between two sets of data samples $\mathbf{X} = (\mathbf{x}_1, \dots \mathbf{x}_n)$ and $\mathbf{Y} = (\mathbf{y}_1, \dots, \mathbf{y}_m)$ can be calculated by summing up the covariances of all the possible combinations of two data samples

$$q(\mathbf{X}, \mathbf{Y}) = \sum_{i=1}^{n} \sum_{j=1}^{m} q(\mathbf{x}_i, \mathbf{y}_j) \tag{2.16}$$

where $q(\mathbf{x}_i, \mathbf{y}_j)$ is calculated using the equation 2.12.

The *canonical correlation*, however, is not just the correlation between two sets of data samples, but the correlation between a *linear combination* of one set and a linear combination of the other set of data samples. Linear combinations of $\mathbf{X}$ and $\mathbf{Y}$ are

$$\mathbf{U} = a_1\mathbf{x}_1 + a_2\mathbf{x}_2 + a_3\mathbf{x}_3 + \cdots + a_n\mathbf{x}_n$$

$$\mathbf{V} = b_1\mathbf{y}_1 + b_2\mathbf{y}_2 + b_3\mathbf{y}_3 + \cdots + b_m\mathbf{y}_m$$

CCA seeks linear combinations $\mathbf{U}$ and $\mathbf{V}$ that have the maximum correlation among all the possible linear combinations of $\mathbf{X}$ and $\mathbf{Y}$. Since $\mathbf{x}_1, \ldots \mathbf{x}_n$ and $\mathbf{y}_1, \ldots, \mathbf{y}_m$ are known, CCA needs to find sets of coefficients $a_1, a_1, \ldots, a_n$ and $b_1, b_2, \ldots, b_m$. Similarly to the equation 2.13, the correlation between linear combinations $\mathbf{U}$ and $\mathbf{V}$ can be calculated with

$$\rho = \frac{q(\mathbf{U}, \mathbf{V})}{\sqrt{q(\mathbf{U}, \mathbf{U})q(\mathbf{V}, \mathbf{V})}} \tag{2.17}$$

This correlation is called canonical correlation. The pair $(\mathbf{U}, \mathbf{V})$ is called the first pair of canonical variates. It is possible to find up to $\min(m, n)$ pairs of canonical variates, but in the method proposed by Lin *et al.* [28] only the first pair of canonical variates is used.

The canonical correlation between the EEG recording and every set of reference signals is calculated in CCA method. Every target has different set of reference signals as already mentioned in the previous section and therefore target whose set of reference signals has the highest canonical correlation with EEG recording will be selected as user's choice. See figure 2.11 for graphical illustration.

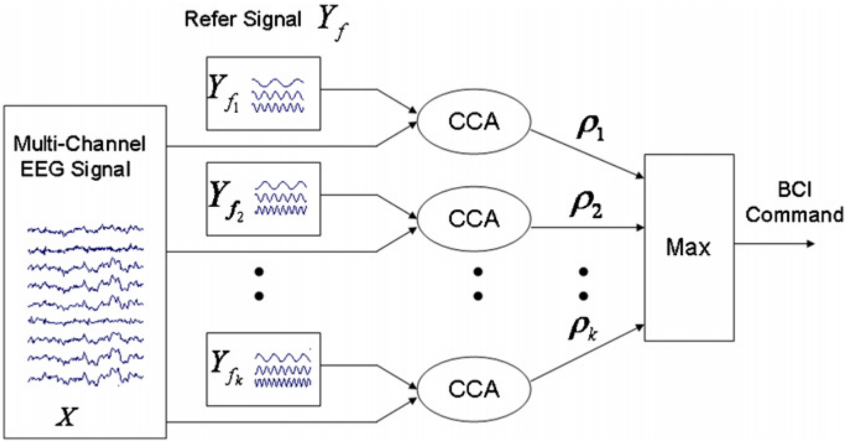


Figure 2.11: The visualisation of the steps of CCA feature extraction method [5].

These two feature extraction methods described in this chapter can be used as separate feature extraction methods, but in the application described in chapter 3, these two methods complement each other and in a sense work as a single feature extraction method.

2.7 Related work

There are many types of BCIs available. A review by Bashashati *et al.* [3] contains a detailed overview of EEG-based BCIs. Their paper includes overview of the following neuromechanisms used in EEG-based BCIs: VEP and SSVEP, P300, slow cortical potential, response to mental task, sensorimotor activity, and multiple neuromechanisms or hybrid BCI. Hybrid BCI uses at least two different neuromechanisms in a BCI and therefore at least two different methods are required to analyse the EEG recording [1, 19].

SSVEP-based BCIs can be divided in categories according to the method used to detect SSVEPs in EEG recording. Current SSVEP-based BCI feature extraction methods include:

- PSDA method introduced by Cheng *et al.* [12].
- Stability coefficient (SC) method introduced by Wu and Yao [45].
- Dual-frequency SSVEP methods [32, 25].
- Multi-phase cycle coding (MPCC) method introduced by Tong *et al.* [40].
- Minimum energy combination (MEC) method introduced by Friman *et al.* [16]. This method has shown better performance than SC and PSDA method [42].
- CCA method introduced by Lin *et al.* [28]. This method has shown better performance than PSDA method [17, 5, 28].
- Multiway CCA method introduced by Zhang *et al.* [51]. This method has shown better performance than standard CCA [51].
- Least absolute shrinkage and selection operator (LASSO) method introduced by Zhang *et al.* [50]. This method has shown better performance than CCA method [50].
- Likelihood ratio test (LRT) method introduced by Zhang *et al.* [49]. This method has shown better performance than CCA method and similar performance to LASSO method [49].

For more comprehensive review of the feature extraction methods see article by Liu *et al.* [29].

In this thesis Emotiv EPOC is used to record brain activity and the recording is analysed using PSDA and CCA feature extraction methods. Table 2.1 gives overview of the existing BCIs that also use Emotiv EPOC for recording brain activity.

	Method	Accuracy (%)	Target detection time (sec)	ITR (bits/min)
Liu *et al.* [30]	CCA	95.83 ± 3.59	5.25 ± 2.14	20.97 ± 0.37
Lin *et al.* [27]	CCA	76.60 ± 21.74	4.34 ± 0.08	14.38 ± 9.04
Choi and Jo [7]	CCA	84.4 ± 5.0	N/A	11.6 ± 3.9
	ERD	84.6 ± 5.3	N/A	11.8 ± 3.7
	P300	89.5	5.35	18.1 ± 2.1
Hvaring and Ulltveit-Moe [24]	CCA	85.12 ± 4.58	3.00	32.92 ± 4.72
	PSDA	89.29 ± 6.41	5.14 ± 0.98	23.78 ± 7.15

Table 2.1: Comparison of the existing BCIs that use Emotiv EPOC.

It can be seen that existing applications have target detection time above or exactly three seconds which is not fast enough to control a robot in real time. However, as Zier mentioned in his paper [53] and as also noticed in the process of developing this application, if the window length is long enough for the feature extraction method to accurately detect correct target, the target detection time is about half the window length. That is so, because after half the window length, half of the previous data is replaced with new data and if there is more new data than previous, then the BCI recognises the new user's choice.

This thesis aims at developing BCI with faster target detection time than the previous applications to provide suitable speed for controlling a robot.

3 Application for controlling a robot via SSVEP

The aim of this chapter is to describe the SSVEP-based BCI designed as a practical part of this thesis. The BCI is written in Python 2.7 and the code is accessible from Github repository, see appendix IV for details. The BCI requires only Emotiv EPOC headset and a computer with Windows operating system, no specific hardware like digital signal processors or LEDs are used.

3.1 Controlling a robot with the application

This section describes how the BCI can be used to control a robot. Each target in the BCI can be used as one command for the robot. The robot can be controlled by looking at different targets on the computer screen in certain sequence, depending on which command should be given to the robot. Then the BCI tries to identify which target is being looked at and translates the results into commands for the robot. The targets could be designed for example as shown in figure 3.1.

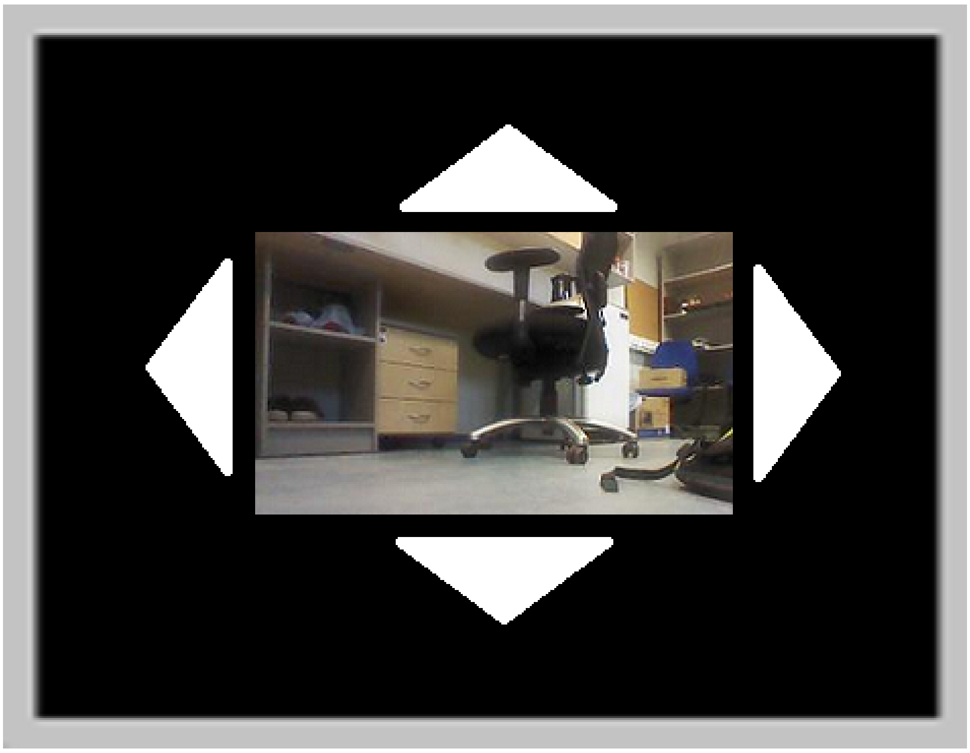


Figure 3.1: Stimuli locations on the screen for controlling a robot. The picture in the middle represents camera's video stream.

The robot[1] used for testing the application has five possible commands: move forward, move backward, turn left, turn right and stop. The robot can execute one command at a

[1] https://github.com/kuz/Garage48-PiTank

time, meaning that it is not possible to move forward and turn left or right at the same time. If the robot is moving forward and receives a command to turn left, the moving forward will stop and the robot starts to turn left. See figure 3.2 for a picture of the robot used for testing the application.

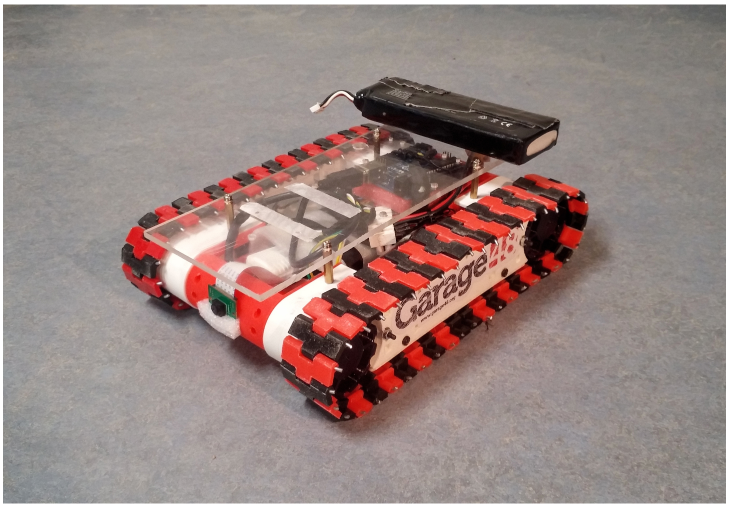


Figure 3.2: The robot used for testing the application.

The robot has a camera on it and thus it is possible to see the camera's videos stream on the computer screen when controlling the robot with the application. The picture in the middle of the screen as shown in figure 3.1 represents the video stream.

The flowchart of the BCI can be seen in figure 3.3. The brain and the Emotiv EPOC headset were discussed in the first chapter. The targets and feature extraction methods of the BCI along with the signal processing techniques were discussed in the second chapter. The details of the actual implementation of the signal pipeline and the novel target identification method will be discussed in sections 3.3 and 3.4 respectively.

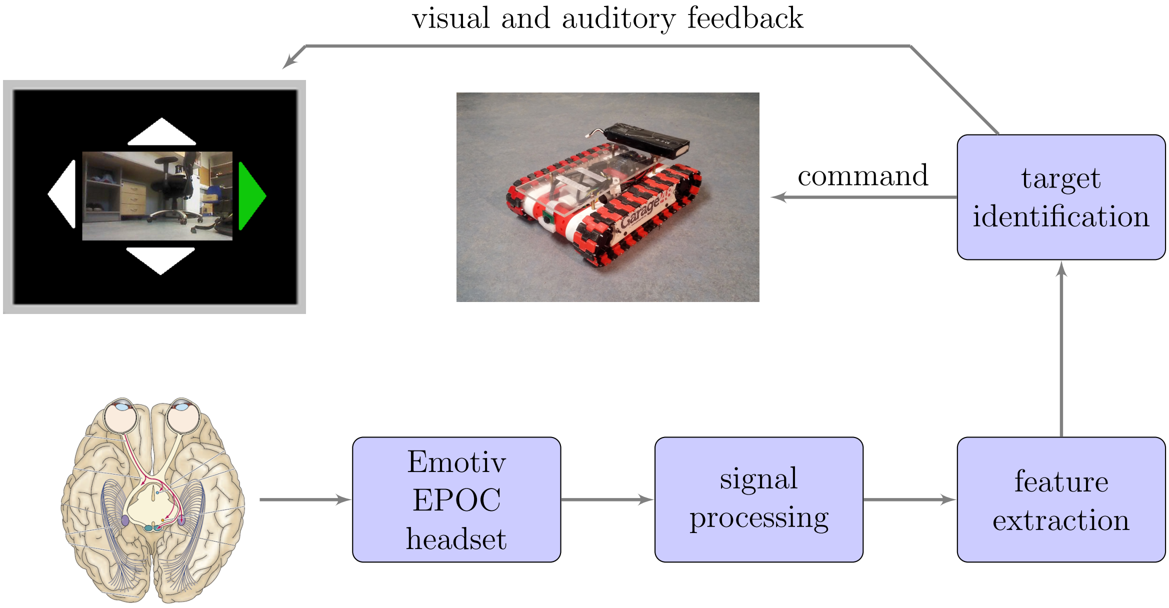


Figure 3.3: Flowchart of the BCI.

The robot used in testing the application has very common commands—forward, back, left, right and stop. Therefore, if the BCI proves to have sufficiently high target detection speed and accuracy, then it could be adapted to other devices that can be controlled with similar commands.

3.2 Overview of the application

As already mentioned, the application implements a communication channel between the brain and a robot, thus making it possible to send control commands to the robot. Brain activity is recorded using Emotiv EPOC and the recording is analysed with PSDA and CCA feature extraction methods. The application is open-source, written in Python 2.7 and requires Windows operating system.

The application is divided into the following components:

- `MainWindow` is the window through which a user can easily change the parameters of the signal pipeline, feature extraction methods and targets and control the flow of the application through graphical user interface (GUI);
- `TargetsWindow` is the window on which the targets are displayed;
- `Emotiv` is the class which receives the raw data from the Emotiv EPOC headset, decrypts the data and sends it to other components of the application;
- `Robot` is the code that sends commands to the robot after the signal has been analysed and result has been obtained;
- `Plot` is the code that can be used to plot the EEG signal in real-time;
- `Extraction` is the code that analyses the signal using PSDA and CCA feature extraction methods.
- `PostOffice` is the class that handles the communication between the components of the application.

There is instance one of each component except the `Plot` and the `Extraction`. The number of `Plots` and `Extractions` depends on how many plots the user wants to see or how many feature extraction methods to use. Each plot and feature extraction method can have different parameters. Having different parameters on different plots allows the user to compare how different signal processing affects the signal. Having different parameters on feature extraction methods allows the user to compare how the parameters affect the performance of the BCI. Furthermore, feature extraction methods with different parameters can be used together to complement each other and work as a single feature extraction method. Combining different feature extraction methods to a single method is further discussed in section 3.4. For graphical illustration of the components of the application see figure 3.4.

The `PostOffice` class is important because each component is running on separate subprocess, meaning that each component has its own memory space. Thus it is not possible for the components of the application to communicate by for example writing to and

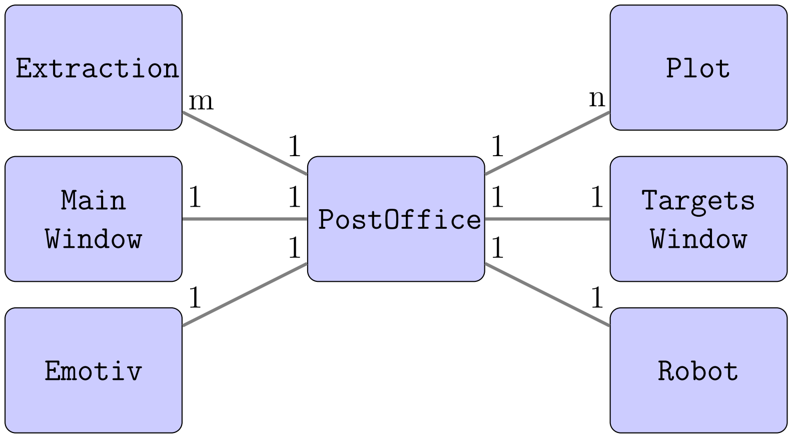


Figure 3.4: Components of the application. Lines between the components represent connections through which the components communicate. The numbers show how many instances of the same component there can be.

reading from the same variable—their memory is separated—and therefore more complex type of communication has to be used. Alternative to using multiple processes or multiprocessing is using multiple threads or multithreading. Unlike processes, threads run in the same memory space but due to the limitations of the standard implementation of Python, threads cannot use multiple central processing units (CPUs) or multiple CPU cores at the same time while processes can.

Making use of multiple CPU cores is important because calculating the power spectral density and analysing the recorded signal using CCA is computationally expensive and thus it might be necessary to divide the calculations between different CPU cores to achieve optimal performance and use the BCI in real-time. The Emotiv EPOC headset is constantly sending new data to the application and the application has to be able to finish analysing current data before receiving enough data for the next feature extraction.

For the ease of use, the application has GUI. See figure 3.6 for an example of the application's user interface. Multiprocessing is also important to avoid freezing of the whole application while one window of the GUI is moved or resized. In case of moving a window, the window manager of Windows operating system freezes the subprocess which is updating the window. If the application is divided into subprocesses then only the subprocess which updates the window freezes, the rest of the application continues working.

The usage of multiple subprocesses is achieved by using multiprocessing module from Python 2.7 standard library. This module also provides connections that can be used for communication between the components of the application. In addition to the Python 2.7 standard library, the following libraries were used:

- Emokit[2] to access raw data from Emotiv EPOC headset;
- PsychoPy [35] for designing visual stimuli and precise timing of the stimuli presentations;
- scikit-learn [34] for calculating CCA algorithm;

[2] https://github.com/openyou/emokit/tree/master/python/emokit

- SciPy and NumPy [36] for calculating other advanced mathematical algorithms, for example FFT;
- PyQtGraph[3] for real-time plotting of the data.

The most important part of the implementation of this application is that multiple feature extraction methods complement each other when analysing EEG recording.

3.3 Signal pipeline of the application

As already discussed in the previous section, the application does not have one specific signal pipeline, but the parameters of the signal pipeline can be changed from the GUI. This allows different configurations to be easily tested to find the best settings for controlling a robot and the best settings for different users. See figure 3.5 for the flowchart of the application's signal pipeline.

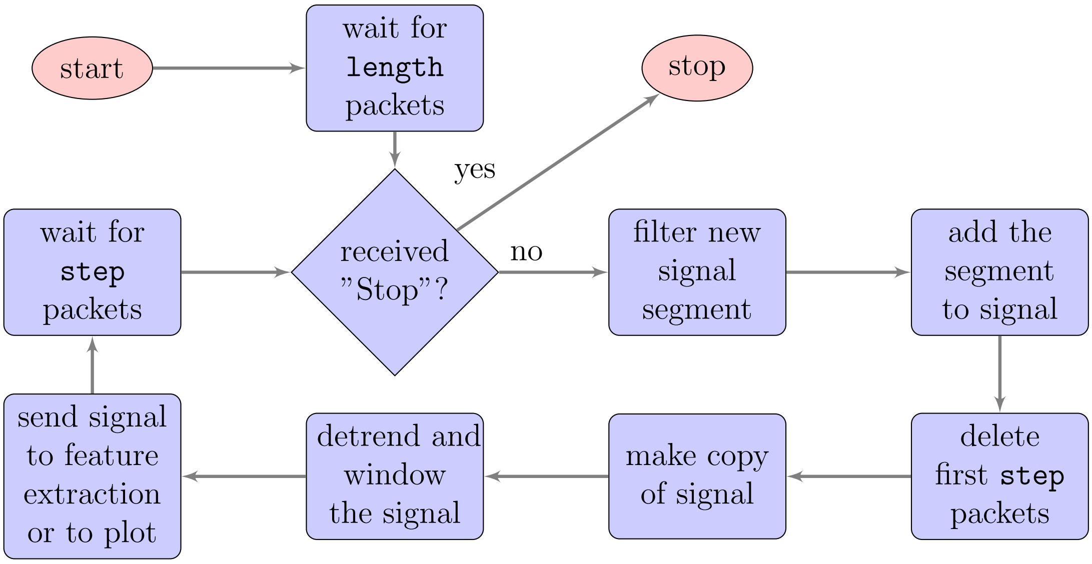

start
wait for length packets
stop
yes
received "Stop"?
no
wait for step packets
filter new signal segment
add the segment to signal
send signal to feature extraction or to plot
detrend and window the signal
make copy of signal
delete first step packets

Figure 3.5: Flowchart of the application's signal pipeline.

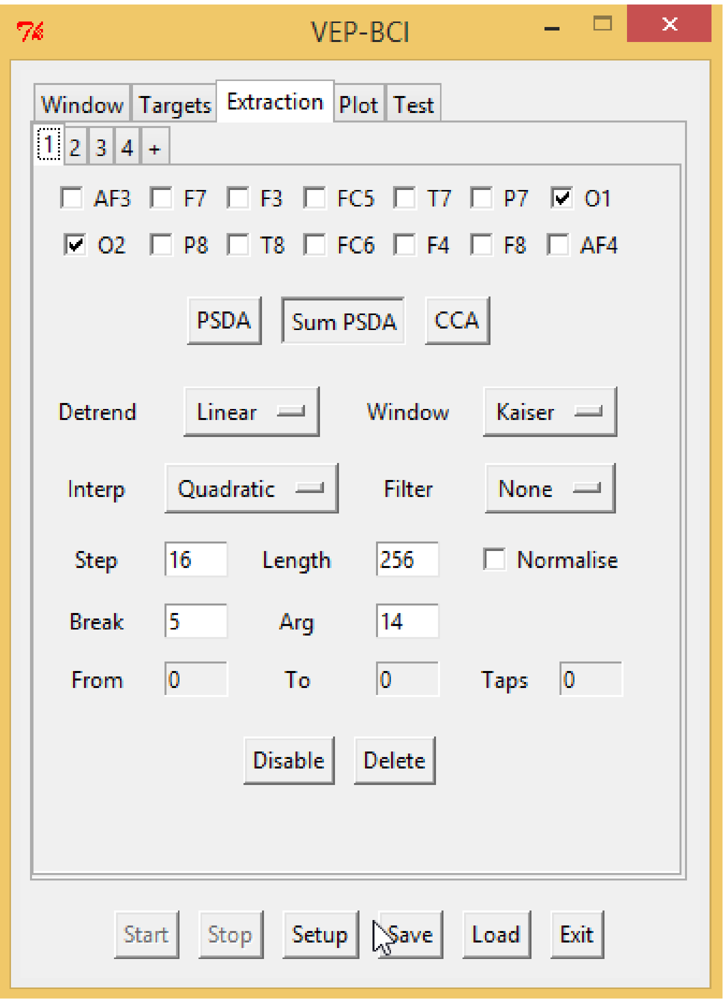
As seen in figure 3.5, `length` is the window length and it shows how long filtered signal is held in memory. `Step` on the other hand shows how many packets have to be received before trying to identify user's choice. If `step` is smaller than `length` then the window is overlapping which means that some of the data that was used in previous feature extraction will also be used in the next feature extraction. If `step` is equal to `length`, then every consecutive feature extraction uses only new data that has not been analysed by previous feature extractions. It would not make sense to use larger `step` than `length`, because then some of the data will not be analysed and in this case the target detection would be slower.

After receiving `step` packets, the whole filtered signal will be first detrended and then windowed. Detrending and windowing were thoroughly discussed in sections 2.5.1 and 2.5.2 respectively. Since some data is deleted and new data is added to the signal after

[3] http://www.pyqtgraph.org

every step, the windowing cannot be performed only on the previously received signal segment, but has to be performed on the whole signal every time. Detrending could be performed only on the previously received signal segment, but using it on the whole signal gives more possibilities since the same result as would be obtained by detrending every new signal segment can be achieved by dividing the detrending of the whole signal into segments with suitable length. In the application this division can be achieved by giving the `break` parameter which shows how many breakpoints to use that divide the signal into equal length segments for detrending.

There are more signal pipeline settings that can be changed through the GUI of the application and now a brief overview of these settings will be given. See figure 3.6 for illustration of the application's GUI for choosing signal pipeline components and parameters.



☐ AF3 ☐ F7 ☐ F3 ☐ FC5 ☐ T7 ☐ P7 ☑ O1

☑ O2 ☐ P8 ☐ T8 ☐ FC6 ☐ F4 ☐ F8 ☐ AF4

Detrend Linear Window Kaiser

Interp Quadratic Filter None

Step 16 Length 256 ☐ Normalise

Break 5 Arg 14

From 0 To 0 Taps 0

Figure 3.6: The user interface for choosing signal pipeline options.

The explanation of the signal pipeline settings is the following:

1. `Detrend` is either linear, constant or none. None means that detrending is not used. Linear means that linear trend is removed from the raw signal; constant means that constant trend is removed from the raw signal. Detrending was more thoroughly discussed in section 2.5.1.

2. `Window` is either hann, hamming, blackman, keiser, bartlett or none. None means that window function is not used. Other options are standard signal processing window functions. Windowing was more thoroughly discussed in section 2.5.2.

3. `Interp` is either linear, nearest, zero, slinear, quadratic or cubic. See SciPy documentation[4] for more information. Interpolation was more thoroughly discussed in section 2.5.3.

4. `Filter` is either high-pass, low-pass, band-pass or none. None means that filtering is not used. High-pass filter means that frequencies lower than the given value are removed from the signal. Similarly, low-pass filter removes frequencies higher than the given value. Band-pass filter takes two values and removes frequencies that are not in the given range. See SciPy documentation[5] for more information.

5. `Step` shows how many packets have to be received before trying to identify user's choice. For example, if sampling rate is 128 Hz and step is 64 then the feature extraction algorithms are executed after every $\frac{64}{128\,\mathrm{Hz}} = 0.5$ seconds.

6. `Length` is the length of the window. Length shows the number of packets on which the feature extraction algorithms are executed. For example, if sampling rate is 128 Hz and length is 512 then the feature extraction algorithms are performed on the last $\frac{512}{128\,\mathrm{Hz}} = 4$ seconds of data.

7. `Break` is the number of breakpoints used when detrending the signal. The breakpoints will be equally spaced. If the number of breakpoints is 1, then the breakpoint will be in the middle of the signal and the trend will be removed separately from the first half of the signal and the second half of the signal.

8. `Arg` is the beta argument for kaiser window. See NumPy documentation for details[6].

9. `Normalise` shows whether to normalise the estimated power spectral density as in equation 2.9 or not.

10. `From` and `to` are the frequencies used to specify which frequencies are removed and which not. From shows the lowest frequency that is passed, lower frequencies than the value will be removed; to shows the highest frequency that is passed, higher frequencies than the value will be removed.

11. `Taps` shows the number of taps or the length of the filter. See SciPy documentation[5] for details.

Since the application has GUI and many parameters that can be easily tested, this application can be a good tool to compare how different signal pipelines affect the detection of SSVEPs.

[4] https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.interp1d.html

[5] http://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.firwin.html

[6] http://docs.scipy.org/doc/numpy/reference/generated/numpy.kaiser.html

3.4 Target identification method

This section describes the target identification method used in this application. As is the case with the signal pipeline, this application actually does not have one specific feature extraction method, but the feature extraction method can be changed and a combinations of different feature extraction methods can be used. To the best of the author's knowledge, combining different feature extraction methods while using only SSVEP neuromechanism has not been used before.

Currently the application has three different feature extraction methods that can be combined: widely known PSDA and CCA method and a method that is similar to the PSDA method but the estimated power spectral density is calculated not for each signal obtained from different channels but for the sum of all the signals from different channels. If only one channel is used, then this method works exactly the same as standard PSDA method. In the GUI, this method is called `sum PSDA` method. Using this method makes sense, because FFT is linear, meaning that the result will be the same if first the signals are summed up and then the power spectral density is estimated and if first the power spectral density is estimated separately for each channel and then the power spectral density estimates for each channel are summed up.

Unlike the sum PSDA method, the standard PSDA method calculates separate results for each channel. Both methods send the results to the `PostOffice` separately for each harmonic. Which harmonics are used for the detection of which target can be changed in the GUI. It is also possible to get the result for the sum of different harmonics. The CCA method is different from the PSDA methods as it gives always only one result, even if multiple channels and multiple harmonics are used.

Since multiple feature extraction methods can be used together and some feature extraction methods give more than one result, each feature extraction result is given a certain weight which shows how much this result affects the final decision. For example, if CCA and sum PSDA method are used together and from the sum PSDA results the result of the first harmonic and the result of the sum of all used harmonics are used for each target, then the weights could be 1 for each sum PSDA result and 2 for CCA result. Then the target could be finally chosen for example if the sum of the weights for a target is at least 3. This means actually that target is chosen only if CCA method chooses a target and at least one of the sum PSDA results is the same. This case could be easily implemented without using weights, but when using more feature extraction methods, the conditions for choosing a target without using the weights will get more complicated and thus it is easier to use the weights to make the final decision.

To further improve the performance of the BCI, not only the weights of the last results, but the weights of multiple previous results are used to make the decision. Thus the condition would not be that if the sum of the weights of the last results is at least 3, but for example the sum of the weights of the three previous results is at least 9. If there are multiple targets that satisfy the condition, then the target with higher sum of the weights of the previous results is chosen. This case is not possible in the previously given example, since all the results always correspond to one target. In the previous example the maximum sum of weights for one set of results is 4, therefore the sum of weights for the last three sets of results is $4 \cdot 3 = 12$. If the condition is that this sum has to be higher than 9, then the maximum sum of weights for another target is $12 - 9 = 3$.

Finally, the target is chosen if n out of m previous results that were obtained by comparing the sum the weights of k previous results are the same. See figure 3.7 for graphical illustration of the target identification method.

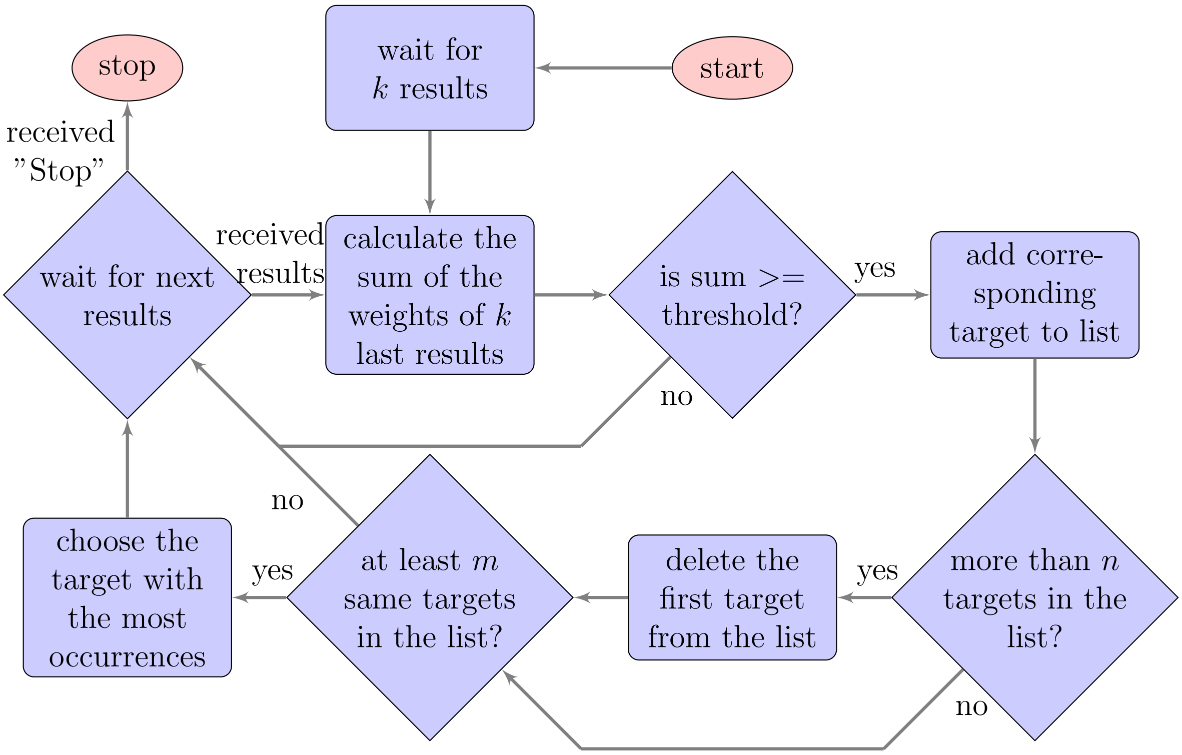


Figure 3.7: Flowchart of the application's target identification.

It is easy to modify target identification method by changing the code in `handleFreqMessage` method in `PostOffice.py`. This method receives the results of the feature extraction methods through a connection similarly to the data accessing code discussed in section 3.5. The results have already been organised into different data structures.

It is also worth mentioning that different feature extraction methods can have different signal pipelines. For example it would not make sense to window a signal that is later analysed using CCA, but windowing could be beneficial if the power spectral density of the signal is later estimated analysed using PSDA method. Thus the application has the functionality to use multiple signal pipelines with different options, which gives even more flexibility.

Using PSDA and CCA method together makes the target identification more accurate, because finally the target is chosen only if different methods give the same result. If one of the methods makes a mistake and gives a wrong result, then the other method might still give the correct result. It is less probable for two methods to make the same mistake at the same time than for one method to make a mistake. Using these two methods together makes sense, because these methods analyse the EEG signal from two different aspects—one uses time-domain representation, the other uses frequency-domain representation.

To further improve this method, the phase information from the frequency spectrum that is currently being discarded could also be used. One possible usage would be to design CCA method reference signals with correct phase. Currently both sine and cosine waves are used to achieve optimal minimum correlation as discussed in section 2.6.2. But even without this improvement, combining these methods already improves the performance of the BCI.

3.5 Using the application with other EEG devices

Although the application was written for Emotiv EPOC, it can be used with other EEG devices too if it is possible to access the data of these devices with Python script. As discussed in section 3.2, the communication between the data accessing code and the rest of the application is implemented using Python multiprocessing module. The data accessing code runs on separate process and communicates with the rest of the application by sending messages through multiprocessing connection. It sends raw data through the connection and receives messages like ”Setup”, ”Start”, ”Stop” and ”Exit”. Each of the received messages should lead to calling a specific function. If the received message is

- ”Setup”, then the object should get internally ready to start sending data;
- ”Start”, then the object should start sending the data;
- ”Stop”, then the object should stop sending the data;
- ”Exit”, then all the needed clean up procedures should be executed, because the application is closing.

The easiest way to start using different headset with the application is changing the code in `MyEmotiv.py`. The constructor of `MyEmotiv` class has to take one argument. This argument is the object that handles the communication between different processes. The last line in the constructor of `MyEmotiv` should call the argument's `waitMessages` method, which waits until it receives one of the previously discussed messages. The `waitMessages` method takes arguments which are the functions that are called when corresponding message is received. By replacing these methods in `MyEmotiv` class, it is possible to use the application with other EEG devices too. Since the application can be relatively easily modified to use other EEG devices, it could be a good tool to test how suitable are different devices for a SSVEP-based BCI.

4 Results

As discussed in chapter 3, the BCI written as a practical part of this thesis has many parameters that can be changed and finding the best combination of different parameters needs thorough testing. Unfortunately, thorough testing did not fit into the scope of this thesis. The testing was performed on only one subject. By using trial and error method, settings that are shown in appendix III were fixed for further testing.

After fixing the signal pipeline parameters as shown in table A1, different target identification parameters were tested. This testing was performed in trials, each trial lasted 60 seconds. Four targets were used with frequencies of 5.45, 6.0, 6.67 and 7.5 Hz. The rest of the target parameters can be seen in table A2. The application chose randomly a target that the subject had to look at and after this randomly chosen target was detected from the EEG recording, the application chose randomly another target that the subject had to look at and so on. New random target was chosen only if the previous had been detected. This continued until 60 seconds had passed. If a detected target was not the target that the application randomly chose, then the result was classified as false positive. If a detected target was the same as the target that application randomly chose, then the result was classified as true positive. Accuracy was calculated by dividing the number of true positives with the total number of results.

The application gave the subject visual feedback about which target had been detected from the EEG recording and showed the current target that the subject has to look at as shown in figure 4.1. Laptop computer with 60 Hz refresh rate 14.0" diagonal LED-backlit HD 16:9 widescreen (1366 x 768) LCD monitor was used for displaying the targets. For recording the brain activity, Emotiv EPOC EEG device was used.

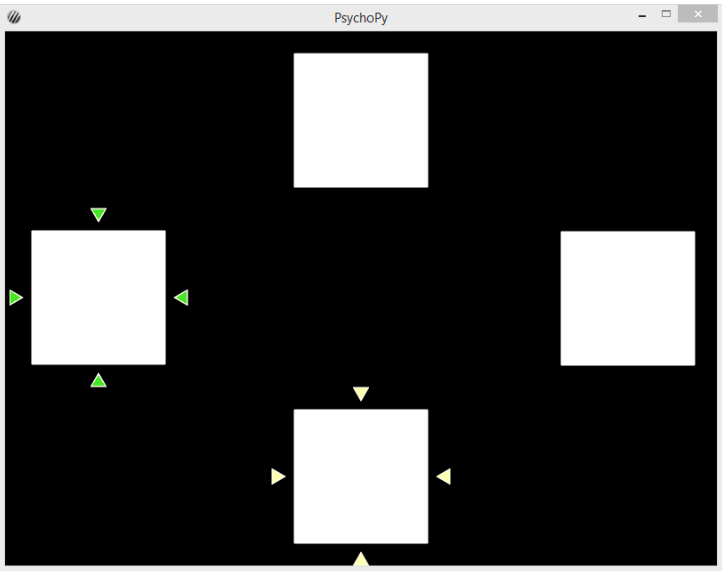


Figure 4.1: The targets used for testing the application and visual feedback for the subject. White triangles show the current target that the subject has to look and the green triangles show the target that was detected.

The feature extraction method's weight parameters were fixed as shown in table 4.1.

The method ID corresponds to the method ID in table A1. Finally, different target identification methods were tested and the results can be seen in figure 4.2. The target detection time was calculated by dividing 60 seconds with the number of results in the given trial. The ITR was calculated as shown in equation 2.4.

Method ID	Sensor	Harmonic	Weight
1	Sum of all	1	6
	Sum of all	Sum of all	6
2	All	All	12
3	O1	1	1
	O2	1	1
	O1	Sum of all	1
	O2	Sum of all	1
	Sum of all	Sum of all	1
	Sum of all	Sum of all	1
4	All	All	6

Table 4.1: Weights of the feature extraction methods.

The threshold, k, m and n in table 4.2 are the target identification parameters that were discussed in section 3.4.

Trial ID	Target detection time (sec)	Accuracy (%)	ITR (bits/min)	Threshold	k	m	n
1	3.00	95.00	32.69	65	3	4	3
2	2.86	90.48	29.30	70	3	3	2
3	2.86	80.95	20.91	65	3	5	3
4	2.86	80.95	20.91	65	3	7	5
5	2.73	95.45	36.55	65	3	5	4
6	2.73	90.91	31.16	65	3	4	3
7	2.73	90.91	31.16	65	3	6	4
8	2.73	81.82	22.61	65	3	8	5
9	2.40	88.00	32.01	65	3	7	4
10	2.40	80.00	24.03	65	3	8	4
11	2.31	84.62	29.56	65	3	3	2
12	2.22	81.48	27.41	60	3	4	3
13	2.14	75.00	22.19	65	3	4	2
Average:	2.61 ± 0.28	85.81 ± 6.39	27.73 ± 5.11				

Table 4.2: Results of 13 trials of testing the application. Threshold is the value that the sum of the weights of the k previous results has to exceed for the corresponding target to be added to the list of results. From the list of results, a target is chosen if there are at least n occurrences of the targets. The list of results has the length of m.

The results presented in this section can be compared to other BCIs that also use Emotiv EPOC headset. The results of the previous BCIs can be seen in table 2.1.

Conclusion

The aim of this thesis has been to describe an SSVEP-based BCI implemented as a practical part of this thesis. In the first chapter the biological background was discussed to understand how brain activity can be measured and a low-cost EEG device called Emotiv EPOC was chosen for recording the brain activity.

The second chapter provided an overview of the mathematical and technological background. Widely used feature extraction methods called PSDA and CCA were discussed in this chapter along with signal processing techniques. All the discussed feature extraction methods and signal processing techniques were used in the implemented application.

Finally, the third chapter described the implemented application itself. The strengths of the application compared to the existing applications are that this application is open-source, it allows different options for signal pipeline to be easily tested and it combines different SSVEP feature extraction methods. Although there are hybrid BCIs that use different neuromechanisms and therefore these BCIs use different feature extraction methods, to the best of the author's knowledge there are no BCIs that use different feature extraction methods with one neuromechanism, in this case SSVEP.

The application could be further developed to support different EEG devices and more feature extraction methods to find the best combination of different feature extraction methods. Using multiple methods at the same time clearly improves the performance of a BCI, but on the other hand it is computationally more expensive. But as the technology improves, this becomes less relevant.

Although the application was not thoroughly tested, it already shows very good results compared to the previous BCIs. Current results show that the application is quite stable with target detection time of 2.61 seconds while the target detection times of the previous BCIs are reported to be 3 seconds or more. Thus it is important to test and improve the application further to find settings that give even better results and maybe some day this BCI will have good enough performance to be adapted for controlling real world devices.

References

[1] S. Amiri, R. Fazel-Rezai, and V. Asadpour. A review of hybrid brain-computer interface systems. *Advances in Human-Computer Interaction*, 2013, 2013.

[2] F. A. Azevedo, L. R. Carvalho, L. T. Grinberg, J. M. Farfel, R. E. Ferretti, R. E. Leite, W. J. Filho, R. Lent, and S. Herculano-Houzel. Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain. *The Journal of Comparative Neurology*, 513(5):532–541, 2009.

[3] A. Bashashati, M. Fatourechi, R. K. Ward, and G. E. Birch. A survey of signal processing algorithms in brain-computer interfaces based on electrical brain signals. *Journal of Neural Engineering*, 4(2):R32–R57, 2007.

[4] G. Bin, X. Gao, Y. Wang, B. Hong, and S. Gao. VEP-based brain-computer interfaces: Time, frequency, and code modulations. *Computational Intelligence Magazine, IEEE*, 4(4):22–26, 2009.

[5] G. Bin, X. Gao, Z. Yan, B. Hong, and S. Gao. An online multi-channel SSVEP-based brain-computer interface using a canonical correlation analysis method. *Journal of Neural Engineering*, 6(4), 2009.

[6] Blausen.com staff. Blausen gallery 2014. *Wikiversity Journal of Medicine*, 1(2), 2014.

[7] S. J. Bongjae Choi. A low-cost EEG system-based hybrid brain-computer interface for humanoid robot navigation and recognition. *PLoS ONE*, 8(9), 2013.

[8] R. G. Brown and P. Y. Hwang. *Introduction to Random Signals and Applied Kalman Filtering*. John Wiley & Sons, 1997.

[9] G. Buzsáki, C. A. Anastassiou, and C. Koch. The origin of extracellular fields and currents–EEG, ECoG, LFP and spikes. *Nature Reviews Neuroscience*, 13(6):407–420, 2012.

[10] N. A. Campell, J. B. Reece, M. R. Taylor, E. J. Simon, and J. L. Dickey. *Biology: Concepts and Connection*. Benjamin Cummings, 6 edition, 2008.

[11] H. Cecotti, I. Volosyak, and A. Graser. Reliable visual stimuli on LCD screens for SSVEP based BCI. *The 2010 European Signal Processing Conference (EUSIPCO-2010)*, page 5, 2010.

[12] M. Cheng, X. Gao, S. Gao, and D. Xu. Design and implementation of a brain-computer interface with high transfer rates. *Biomedical Engineering, IEEE Transactions on*, 49(10):1181–1186, 2002.

[13] A. M. Dale and E. Halgren. Spatiotemporal mapping of brain activity by integration of multiple imaging modalities. *Current Opinion in Neurobiology*, 11(2):202—-208, 2001.

[14] M. Duvinage, T. Castermans, M. Petieau, T. Hoellinger, G. Cheron, and T. Dutoit. Performance of the Emotiv EPOC headset for P300-based applications. *BioMedical Engineering OnLine*, 12(56), 2013.

[15] O. Friman, J. Cedefamn, P. Lundberg, M. Borga, and H. Knutsson. Detection of neural activity in functional MRI using canonical correlation analysis. *Magnetic Resonance in Medicine*, 45(2):323–330, 2001.

[16] O. Friman, I. Volosyak, and A. Graser. Multiple channel detection of steady-state visual evoked potentials for brain-computer interfaces. *Biomedical Engineering, IEEE Transactions on*, 54:742–750, 2007.

[17] G. Hakvoort, B. Reuderink, and M. Obbink. Comparison of PSDA and CCA detection methods in a SSVEP-based BCI-system. *Technical Report TR-CTIT-11-03*, 2011.

[18] W. M. Hartmann. *Signals, Sound, and Sensation.* Springer Science & Business Media, 1997.

[19] J. Henshaw, W. Liu, and D. Romano. Problem solving using hybrid brain-computer interface methods: A review. *Cognitive Infocommunications (CogInfoCom), 2014 5th IEEE Conference on*, pages 215–219, 2014.

[20] C. S. Herrmann. Human EEG responses to 1–100 Hz flicker: resonance phenomena in visual cortex and their potential correlation to cognitive phenomena. *Experimental Brain Research*, 137(3-4):346–353, 2001.

[21] M. Hoshiyama and R. Kakigi. Effects of attention on pattern-reversal visual evoked potentials: Foveal field stimulation versus peripheral field stimulation. *Brain Topography*, 13(4):293–298, 2001.

[22] H. Hotelling. Relations between two sets of variates. *Biometrika*, 28(3-4):321—-377, 1936.

[23] D. Hubel. *Eye, Brain, and Vision.* Scientific American Library Series. Henry Holt and Company, 1995.

[24] F. T. Hvaring and A. H. Ulltveit-Moe. A comparison of visual evoked potential (VEP)-based methods for the low-cost Emotiv EPOC neuroheadset. Master's thesis, Norwegian University of Science and Technology, 2014.

[25] H.-J. Hwanga, D. H. Kimb, C.-H. Hana, and C.-H. Ima. A new dual-frequency stimulation method to increase the number of visual stimuli for multi-class SSVEP-based brain–computer interface (BCI). *Brain Research*, 1515:66—77, 2013.

[26] J. D. Kropotov. *Quantitative EEG, Event-Related Potentials and Neurotherapy.* Academic Press, San Diego, 2009.

[27] Y.-P. Lin, Y. Wang, and T.-P. Jung. Assessing the feasibility of online SSVEP decoding in human walking using a consumer EEG headset. *NeuroEngineering and Rehabilitation*, 11(119), 2014.

[28] Z. Lin, C. Zhang, W. Wu, and X. Gao. Frequency recognition based on canonical correlation analysis for SSVEP-based BCIs. *Biomedical Engineering*, 54(6), 2007.

[29] Q. Liu, K. Chen, Q. Ai, and S. Q. Xie. Review: Recent developement of signal processing algorithms for SSVEP-based brain computer interfaces. *Journal of Medical and Biological Engineering*, 34(4):299–309, 2014.

[30] Y. Liu, X. Jiang, T. Cao, F. Wan, P. U. Mak, P.-I. Mak, and M. I. Vai. Implementation of SSVEP based BCI with Emotiv EPOC. *Virtual Environments Human-Computer Interfaces and Measurement Systems (VECIMS), 2012 IEEE International Conference*, pages 34–37, 2012.

[31] G. R. Müller-Putz, R. Scherer, C. Brauneis, and G. Pfurtscheller. Steady-state visual evoked potential (SSVEP)-based communication: impact of harmonic frequency components. *Journal of Neural Engineering*, 2(4):123–130, 2005.

[32] T. M. S. Mukesh, V. Jaganathan, and M. R. Reddy. A novel multiple frequency stimulation method for steady state VEP based brain computer interfaces. *Physiological Measurement*, 27(1):61–71, 2006.

[33] P. Olejniczak. Neurophysiologic basis of EEG. *Journal of Clinical Neurophysiology*, 23(3):186–189, 2006.

[34] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[35] J. W. Peirce. Generating stimuli for neuroscience using psychopy. *Frontiers in Neuroinformatics*, 2(10), 2009.

[36] F. Pérez and B. E. Granger. Python for scientific computing. *Computing in Science & Engineering*, 9(3):10–20, 2007.

[37] D. Purves, G. J. Augustine, D. Fitzpatrick, W. C. Hall, A.-S. Lamantia, J. O. McNamara, and S. M. Williams, editors. *Neuroscience.* Sinauer Associates Inc, 3 edition, 2004.

[38] F. Scholkmann, S. Kleiser, A. J. Metz, R. Zimmermann, J. M. Pavia, U. Wolf, and M. Wolf. A review on continuous wave functional near-infrared spectroscopy and imaging instrumentation and methodology. *NeuroImage*, 85:6–27, 2014.

[39] F. Teng, Y. Chen, A. M. Choong, S. Gustafson, C. Reichley, P. Lawhead, and D. Waddell1. Square or sine: Finding a waveform with high success rate of eliciting SSVEP. *Computational Intelligence and Neuroscience*, 2011, 2011.

[40] J. Tong and D. Zhu. Multi-phase cycle coding for SSVEP based brain-computer interfaces. *BioMedical Engineering*, 14(5), 2015.

[41] *Alzheimer's Disease: Unraveling the Mystery.* U. S. Department of Health and Human Services, National Institutes of Health, 2002. (NIH Publication No. 02-3782).

[42] I. Volosyak. SSVEP-based Bremen-BCI interface–boosting information transfer rates. *Journal of Neural Engineering*, 8(3), 2011.

[43] Y. Wang, Y.-T. Wang, and T.-P. Jung. Visual stimulus design for high-rate ssvep bci. *Electronics Letters*, 46(15):1057–1058, 2010.

[44] J. R. Wolpaw, H. Ramoser, D. J. McFarland, and G. Pfurtscheller. EEG-based communication: Improved accuracy by response verification. *IEEE Transactions on Rehabilitation Engineering*, 6(3):326–333, 1998.

[45] Z. Wu and D. Yao. Frequency detection with stability coefficient for steady-state visual evoked potential (SSVEP)-based BCIs. *Journal of Neural Engineering*, 5(1):36–43, 2008.

[46] W. Yijun, W. Ruiping, G. Xiaorong, and G. Shangkai. Brain-computer interface based on the high-frequency steady-state visual evoked potential. *First International Conference on Neural Interface and Control Proceedings*, 2005:37–39, 2005.

[47] P. Yuan, X. Gao, B. Allison, Y. Wang, G. Bin, and S. Gao. A study of the existing problems of estimating the information transfer rate in online brain–computer interfaces. *Journal of Neural Engineering*, 10(2), 2013.

[48] A. Zani, A. M. Proverbio, and M. I. Posner, editors. *The Cognitive Electrophysiology of Mind and Brain*. Academic Press, San Diego, 2003.

[49] Y. Zhang, L. Dong, R. Zhang, D. Yao, Y. Zhang, and P. Xu. An efficient frequency recognition method based on likelihood ratio test for SSVEP-Based BCI. *Computational and Mathematical Methods in Medicine*, 2014, 2014.

[50] Y. Zhang, J. Jin, X. Qing, B. Wang, and X. Wang. Lasso based stimulus frequency recognition model for ssvep bcis. *Biomedical Signal Processing and Control*, 7(2):104—-111, 2012.

[51] Y. Zhang, G. Zhou, Q. Zhao, A. Onishi, J. Jin, X. Wang, and A. Cichocki. Multiway canonical correlation analysis for frequency components recognition in ssvep-based bcis. In B.-L. Lu, L. Zhang, and J. Kwok, editors, *Neural Information Processing*, volume 7062 of *Lecture Notes in Computer Science*, pages 287–295. Springer Berlin Heidelberg, 2011.

[52] D. Zhu, J. Bieger, G. G. Molina, and R. M. Aarts. A survey of stimulation methods used in SSVEP-based BCIs. *Computational Intelligence and Neuroscience*, 2010:1–13, 2010.

[53] B. J. Zier. SSVEP-based brain computer interface using the Emotiv EPOC. Master's thesis, Eastern Washington University, 2012.

Internet URLs were valid on 14.05.2015

Appendices

I Glossary

action potential – event of sending signals used by neurons. 8, 9

axon – nerve fibre through which a neuron sends signals. 7, 8

canonical correlation – correlation between the linear combinations of two data samples. 31

central visual field – very centre of gaze that is clearly seen, not like peripheral vision that is blurry. 12

correlation – the Pearson product-moment correlation coefficient. Covariance divided by the product of standard deviations. 29–31

covariance – measure of similarity. 28–30

cross-correlation – measure of similarity for two signals as a function of similarity versus time lag between the signals. 29, 30

current dipole – formed by a sink and a source around postsynaptic cell. Dipoles produce electric field that can be measured from the scalp using EEG. 9

dendrite – nerve ending through which a neuron receives signals. 7, 8

detrend – removing the trend from a signal. 22–24, 26, 38–40

digital signal – a sequence of values that were obtained from measuring the value of analogue signal in a certain time interval. 13, 18, 19, 22, 25, 26

duty cycle – the percentage of the amount of time the target is in displayed state in one period of the waveform. 16, 20

electromagnetic – functional neuroimaging technique that measures electric or magnetic fields. 9, 10

feature extraction – analysing the EEG recording in a VEP-based BCI. Feature extraction possibly gives multiple results and later the final result has to be chosen from these results. 27–29, 31–33, 35–38, 40–42, 44, 46

flickering – the blinking or the state switches of a target. 15–21

flickering waveform – waveform obtained when plotting the state switches of a target. 16, 17, 19–21

Fourier transform – transforming a signal from time domain to frequency domain. 18, 19, 21

frame – the consecutive images shown on screen that make up the moving picture. 15–17

frequency bin – a value at which the frequency spectrum or the power spectral density function is defined. 19, 23, 25–27

frequency component – a pure tone that a signal contains. 18–22, 25

frequency spectrum – a function of amplitude and phase versus frequency of a signal. 18, 19, 42

functional neuroimaging – measuring an aspect of brain function. 9

fundamental – frequency component of a signal that has the lowest frequency among all the frequency components. 19, 20

haemodynamic – functional neuroimaging technique that measures blood oxygenation and blood flow. 9, 10

harmonic – frequency component of a signal that has a frequency which is integer multiple of the fundamental frequency. 19, 20, 41

interpolation – approximating unknown values. 25, 26, 40

ionophoric protein – proteins used to transport ions across cell membrane. 8

linear combination – for a sequence of values, each value is multiplied by a constant and the result is added up. 31

mean – the measure of central tendency. 22, 24, 26–29

membrane potential – difference in electric potential between the inside of a neuron and the extracellular fluid around the neuron. 8

neural pathway – network formed by functionally related neurons. 8, 12

neuron – nerve cell, main building block of the brain. 7–10

Nyguist frequency – half the sampling rate. 19

pattern reversal – type of a target. Pattern reversal targets reverse their patterns on the screen. 17

periodogram – estimate of power spectral density. 18, 27

peripheral vision – not the centre of gaze but the area that surrounds the centre and is not as clearly seen. 12

postsynaptic cell – a cell that received a signal through a synapse. This term is used for example when discussing single action potential and its effect on the cell that receives the signal. 8

postsynaptic potential – change in the membrane potential in a postsynaptic cell caused by synaptic input. 8, 9

power spectral density – a function of power versus frequency of a signal. 18, 19, 21, 23–28, 37, 40–42

primary visual pathway – neural pathway that is used to convey information that is essential for seeing. 11, 12

pure tone – waveform that contains only one frequency. 17–19, 29

rectangular wave – periodic waveform that is similar to the square wave, but does not have the symmetrical shape, hence its duty cycle is not 50%. The transition between maximum and minimum is instantaneous. 16, 19, 20

reference signal – a signal used in CCA method to find certain frequencies in recorded signal by comparing the recorded signal to it. 29–31, 42

refresh rate – the number of consecutive images shown on screen in one second. 15–17, 44

resting potential – membrane potential of a neuron at a resting state. 8

resting state – state of a neuron that is not exited and not sending signals at the moment. 8

sample – a set of data. 28–31

sampling rate – rate at which samples are obtained for example from an EEG device. 13, 14, 19, 40

single graphic – type of a target. Single graphic targets blink on the screen. 17

sink – part of the current dipole that is more negatively charged than source. 8, 9

source – part of the current dipole that is more positively charged than sink. 8, 9

spectral leakage – if a signal is divided into segments so that a component of the signal does not have an integer multiple of its period in a segment and FFT is performed on this segment, then some of the power of that component's frequency will be distributed over other frequency bins. 23

square wave – periodic waveform that has duty cycle of 50%, is symmetric and the transition between maximum and minimum is instantaneous. 16, 19, 20

state – state of a target. 15–19

statistic – a measure of an attribute of a sample. 28

synapse – a connection between neurons. 7

target – visual stimuli of a VEP-based BCI 15–18, 20, 21, 25, 27–31, 33–36, 38, 41, 42, 44–46

template matching – feature extraction method which compares current signal to previously obtained templates. 29

temporal resolution – the shortest time period of neural activity that is reliably separated out by a measuring technique. 10

threshold potential – certain threshold value. If the membrane potential exceeds threshold potential, action potential is initiated. 8

trend – steady increase or decrease of values in a signal. 21–24

visual processing centre – part of the brain that is responsible for visual perception. 11, 12

window – a function that is used to smooth the ends of a signal. 23–26, 33, 38–40, 42

zero padding – adding zeros to the end of the signal for example to get desired frequency bins in frequency spectrum. 25, 26

II Acronyms

ADC analog-to-digital converter 13, 14

BCI brain-computer interface 2, 15, 18, 20, 21, 24, 25, 27–29, 31–37, 41, 43–46

CCA canonical correlation analysis 2, 6, 27–32, 36, 37, 41, 42, 46

CPU central processing unit 37

EEG electroencephalography 2, 3, 6, 10–15, 18, 19, 21, 23, 25, 27–29, 31, 32, 36, 38, 42–44, 46

ERP event-related potential 11

FFT fast Fourier transform 19, 23, 25, 38, 41

fMRI functional magnetic resonance imaging 9, 10, 28

fNIRS functional near-infrared spectroscopy 9, 10

GUI graphical user interface 36–41

ITR information transfer rate 2, 21, 45

LASSO least absolute shrinkage and selection operator 32

LED light-emitting diode 15, 34

LRT likelihood ratio test 32

MEC minimum energy combination 32

MEG magnetoencephalography 10

MPCC multi-phase cycle coding 32

PET positron emission tomography 9, 10

PSDA power spectrum density analysis 2, 6, 27, 28, 32, 36, 41, 42, 46

SC stability coefficient 32

SNR signal-to-noise ratio 27, 28

SSVEP steady-state visual evoked potential 2, 12–17, 19–23, 25, 27–30, 32, 34, 40, 41, 43, 46

VEP visual evoked potential 11–13, 15, 32

III Fixed parameters for testing

Method ID	Extraction method	Length (sec)	Step (sec)	Sensors	Detrend	Detrend breakpoints	Window	Kaiser beta	Interpolation	Filter
1	Sum PSDA	2	0.125	O1, O2	Linear	5	Kaiser	14	Quadratic	-
2	CCA	2	0.125	O1, O2, P7, P8	Linear	5	-	-	-	-
3	PSDA	1	0.125	O1, O2	Linear	3	Kaiser	14	Quadratic	-
4	CCA	1	0.125	O1, O2, P7, P8	Linear	3	-	-	-	-

Table A1: Settings of the feature extraction method's signal pipelines.

Target ID	Frequency (Hz)	Harmonics in extraction	Position x (pixel)	Position y (pixel)	Color	Type	Shape	Width (pixel)	Height (pixel)
1	5.45	1, 2, 3	0	-200	White	Single graphic	Square	150	150
2	6.0	1, 2, 3	-300	0	White	Single graphic	Square	150	150
3	6.67	1, 2, 3	0	200	White	Single graphic	Square	150	150
4	7.5	1, 2, 3	300	0	White	Single graphic	Square	150	150

Table A2: Settings of the targets.

IV Code of the application

The application written as a practical part of this thesis is open-source and the code is accessible from Github repository[1]. The detailed description of the application is in chapter 3 of this thesis. For the most up to date instructions on which libraries are required, how the repository is structured and how to start using the application, see `README.md` file in the repository.

[1] https://github.com/kahvel/VEP-BCI

Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, **Anti Ingel** (16.02.1993)

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to:

   1.1 reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and

   1.2 make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives until expiry of the term of validity of the copyright,

   of my thesis
   Control a Robot via VEP Using Emotiv EPOC
   supervised by Ilya Kuzovkin

2. I am aware of the fact that the author retains these rights.

3. I certify that granting the non-exclusive licence does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Tartu 14.05.2015