



University of West Bohemia  
Department of Computer Science and Engineering  
Univerzitni 8  
30614 Pilsen  
Czech Republic

# **Methods for Detection of ERP Waveforms in BCI Systems**

The State of the Art and the Concept of Ph.D. Thesis

Tomáš Řondík

Technical Report No. DCSE/TR-2012-10

August, 2012

Distribution: public

Technical Report No. DCSE/TR-2012-10

August 2012

# Methods for Detection of ERP Waveforms in BCI Systems

Tomáš Řondík

---

## Abstract

This thesis summarizes the current state of the art of methods for detection of event-related potential waveforms in brain-computer interface systems. The brain-computer interface helps disabled people to control applications which they are not able to control via a standard user interface due their handicap. The critical part of every brain-computer interface based on event-related potentials is the method which detects the event-related potential waveforms in the input signal.

Nowadays, the detection methods are based on one or more methods from the following domains: statistical methods, methods in time-frequency domain, methods based on decomposition/approximation, artificial neural networks. Representatives of all these domains are described in this thesis.

However, all these methods have weaknesses and it is the opportunity for innovation. Therefore, an innovative method for detection of event-related potential waveforms based on adaptive filtering will be scope of the Ph.D. thesis.

---

Copies of this report are available on <http://www.kiv.zcu.cz/publications/> or by surface mail on request sent to the following address:

University of West Bohemia  
Department of Computer Science and Engineering  
Univerzitni 8  
30614 Pilsen  
Czech Republic

Copyright © 2012 University of West Bohemia, Czech Republic

## Content

1	Introduction .....	4
2	Electroencephalography.....	5
2.1	Origin of the measurable EEG activity .....	5
2.2	EEG characteristics .....	6
2.3	Major EEG rhythms .....	6
2.4	Acquiring and recording of an EEG activity .....	7
2.4.1	Electroencephalograph .....	8
2.4.2	Electroencephalogram .....	8
2.5	Artifacts .....	9
3	Event-related potentials .....	11
3.1	ERP waveform.....	11
3.2	ERPs naming conventions .....	11
3.3	Major ERP components.....	12
3.3.1	Visual sensory responses .....	12
3.3.2	Auditory sensory responses .....	12
3.4	Averaging .....	13
3.5	Analysis of variance for selecting epochs for averaging.....	14
3.5.1	Principle.....	14
3.5.2	Practical use .....	15
3.6	Baseline .....	16
3.7	10 simple rules for designing ERP experiments .....	16
4	Brain-computer interface .....	17
4.1	The parts of an ERP based BCI system.....	18
4.1.1	Stimulator .....	19
4.1.2	Signal acquisition .....	20
4.1.3	Signal preprocessing .....	20
4.1.4	Feature extraction.....	21
4.1.5	Classification .....	21
4.1.6	Extraction of meaning .....	22
4.2	Is everybody able to use an ERP based BCI system? .....	23
5	Methods suitable for ERP waveforms detection .....	25
5.1	Linear discriminant analysis .....	25
5.1.1	ERP detection with LDA.....	26

5.2	Fourier transform .....	26
5.2.1	Continuous Fourier transform.....	26
5.2.2	Discrete Fourier transform.....	27
5.2.3	Short-time Fourier transform .....	27
5.3	Wavelet transform.....	28
5.3.1	Principles of continuous wavelet transform .....	29
5.3.2	Principles of discrete wavelet transform .....	31
5.3.3	ERPs detection with WT .....	33
5.4	Matching pursuit algorithm .....	34
5.4.1	Basic principle .....	34
5.4.2	Output visualization .....	36
5.4.3	ERPs detection with MP algorithm.....	37
5.4.4	ERPs detection issues.....	39
5.5	Hilbert-Huang transform .....	41
5.5.1	Intrinsic mode function.....	42
5.5.2	Empirical mode decomposition .....	42
5.5.3	Stopping criteria.....	43
5.5.4	Hilbert transform.....	44
5.5.5	Standard discrete-time analytic signal.....	44
5.5.6	Determining information about frequencies and amplitudes.....	44
5.5.7	ERPs detection with HHT .....	45
5.6	Artificial neural networks.....	45
5.6.1	Mathematical model of artificial neural networks .....	46
5.6.2	Typical architectures .....	47
5.6.3	Learning.....	47
5.6.4	Artificial neural networks vs. ERP waveforms detection .....	48
5.6.5	Feature vectors .....	49
5.6.6	ART2 neural network.....	50
5.6.7	Self-organizing map (Kohonen map).....	52
6	Scope of the Ph.D. Thesis .....	57
6.1	Disadvantages of introduced methods.....	57
6.2	Adaptive filters.....	58
6.2.1	Adaptive filters and BCI – idea .....	58
7	Conclusion .....	59

7.1 Aims of the Ph.D. thesis .....	59
References.....	60
List of figures .....	64
List of abbreviations .....	66
Appendix A.....	68
Appendix B .....	69

# 1 Introduction

The beginning of electroencephalography (EEG) is dated to year 1929, when Hans Berger reported a brand new set of experiments in which he demonstrated that it is possible to measure the electrical activity of the brain by electrodes placed on the scalp. The signal from electrodes was amplified and the changes in voltage were plotted over time.

A big progress in processing of data obtained by electroencephalography becomes with computer revolution, especially methods for a discrete signal processing. As a result of using methods for discrete signal processing in electroencephalography, a new branch of science was established – the neuroinformatics.

One of the most fascinating parts of research in neuroinformatics is the research of a brain-computer interface (BCI): a system which is able to control an application on the basis of commands distinguished from a brain activity. The greatest importance have the brain-computer interface systems for disabled people who cannot control a specific application via a standard interface.

Nowadays, the major part of brain-computer interfaces is based on event-related potentials (ERP) (especially the P3 potential). From the discrete signal processing point of view, the most challenging task is to recognize whether the signal acquired by electroencephalography contains an ERP waveform or not. There are a few phenomena which make this task nontrivial:

- electromagnetic noise from surrounding environment,
- biological artifacts,
- EEG itself,
- and the fact that ERP waveforms are unique for every one person.

Therefore, design, implementation, and validation of a suitable method for ERP waveforms detection is the key factor for successful implementation of a BCI system as well as one of the main topics on conferences on neuroinformatics.

This thesis presents the state of the art of methods for ERP waveforms detection with a special emphasis on its application in BCI systems. Presented methods cover all mainstream approaches:

- statistical,
- time-frequency,
- approximation/decomposition,
- and artificial neural networks.

Note that we tested all presented methods in our neuroinformatics research group at the Department of Computer Science and Engineering, and we are familiar with them.

The thesis is organized as follows: At the beginning, the brief introduction to electroencephalography and event-related potentials is given. The next chapter deals with brain-computer interfaces based on ERPs – its design and principles are described. Then, basic principles and suitability of methods for ERP waveforms detection from all categories mentioned above are described. At the end, the scope of Ph.D. thesis is given as well as its aims.

## 2 Electroencephalography

### 2.1 Origin of the measurable EEG activity

The central nervous system (CNS) generally consists of nerve cells and glia cells, which are located between neurons. Each nerve cell consists of axons, dendrites, and cell bodies (see Figure 1). Nerve cells respond to stimuli and transmit information over long distances. An axon is a long cylinder, which transmits an electrical impulse. Dendrites are connected to either the axons or dendrites of other cells and receive impulses from other nerves or relay the signals to other nerves. [9]

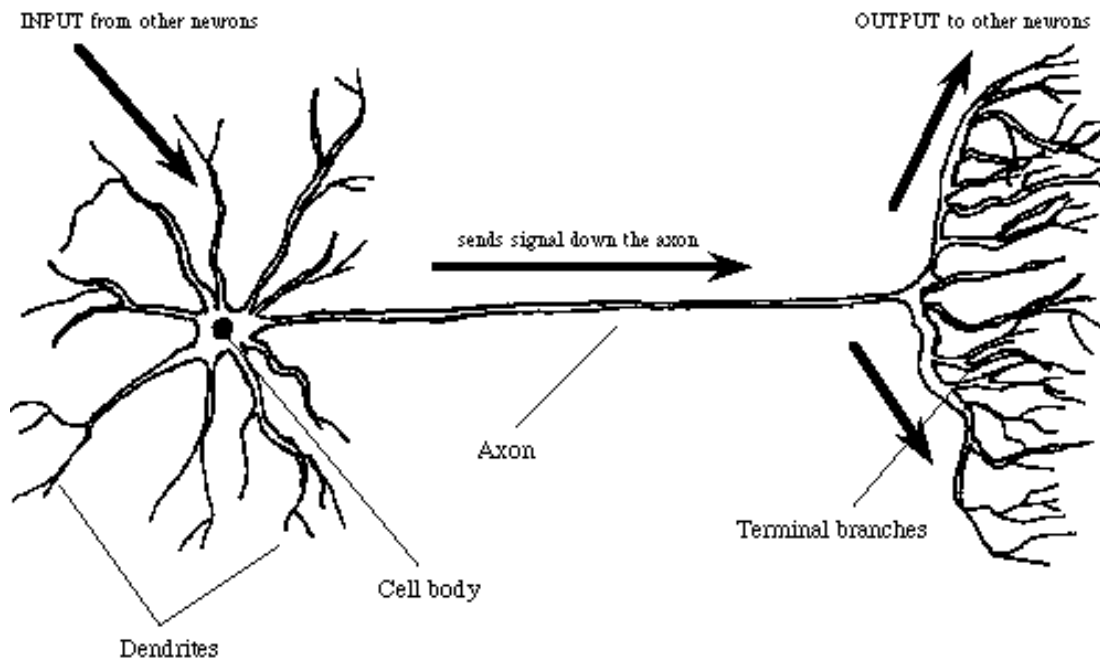


Figure 1: Schema of a nerve cell. [17]

The activities in the CNS are mainly related to the synaptic currents transferred between the junctions (called synapses) of axons and dendrites, or dendrites and dendrites of cells. A potential of 60–70 mV with negative polarity may be recorded under the membrane of the cell body. This potential changes with variations in synaptic activities. If an action potential travels along the fibre, which ends in an excitatory synapse, an excitatory postsynaptic potential (EPSP) occurs in the following neuron. If two action potentials travel along the same fibre over a short distance, there will be a summation of EPSPs producing an action potential on the postsynaptic neuron providing a certain threshold of membrane potential is reached. If the fibre ends in an inhibitory synapse, then hyperpolarization will occur, indicating an inhibitory postsynaptic potential (IPSP). [18; 19; 2]

Following the generation of an IPSP, there is an overflow of cations from the nerve cell or an inflow of anions into the nerve cell. This flow ultimately causes a change in potential along the nerve cell membrane. Primary transmembranous currents generate secondary inonal currents along the cell membranes in the intra- and extracellular space. The portion of these currents that flow through the extracellular space is directly

responsible for the generation of field potentials. Changes in these field potentials, usually with less than 100 Hz frequency, are called EEG activity. [9]

## 2.2 EEG characteristics

Electroencephalography is an electrophysiological method used by doctors and researchers for monitoring of a brain activity. EEG has a very good time resolution by monitoring the potentials, whereas assignment of these potentials to a place of its origin is inaccurate. [20]

In the medicine praxis, the measuring of the EEG is used for [20]:

- examination, whether the patient suffers from epilepsy or migraines, or not
- confirmation or exclusion of the brain death
- determining prognosis for patients in coma
- monitoring of the brain activity during a deep anesthesia
- EEG biofeedback therapy for people (especially young ones) who suffers by learning disabilities, hyperactivity, or impaired concentration

## 2.3 Major EEG rhythms

There are four major brain waves distinguished by their different frequency ranges (see Figure 2). These frequency bands from low to high frequencies respectively are called delta ( $\delta$ ), theta ( $\theta$ ), alpha ( $\alpha$ ), and beta ( $\beta$ ): [9]

1.  $\delta$  wave – frequency from 0.5 Hz to 4 Hz; amplitude usually from 10  $\mu$ V to 300  $\mu$ V. Delta waves are primarily associated with deep sleep and may be present in the waking state. It is very easy to confuse artifact signals caused by the large muscles of the neck and jaw with the genuine delta response. This is because the muscles are near the surface of the skin and produce large signals, whereas the signal that is of interest originates from deep within the brain and is severely attenuated in passing through the skull. [9]
2.  $\theta$  wave – frequency from 4 Hz to 7.5 Hz; amplitude usually more than 20  $\mu$ V. Theta waves appear as consciousness slips towards drowsiness. Theta waves have been associated with access to unconscious material, creative inspiration and deep meditation. A theta wave is often accompanied by other frequencies and seems to be related to the level of arousal. The theta wave plays an important role in infancy and childhood. Larger contingents of theta wave activity in the waking adult are abnormal and are caused by various pathological problems. [9]
3.  $\alpha$  wave – frequency from 8 Hz to 13 Hz; amplitude from 30  $\mu$ V to 50  $\mu$ V. Alpha waves have been thought to indicate both a relaxed awareness without any attention or concentration. The alpha wave is the most prominent rhythm in the whole realm of brain activity and possibly covers a greater range than has been previously accepted. [9]
4.  $\beta$  wave – frequency from 13 Hz to 30 Hz; amplitude usually from 5  $\mu$ V to 30  $\mu$ V. A beta wave is the usual waking rhythm of the brain associated with active thinking, active attention, focus on the outside world, or solving concrete problems, and



is found in normal adults. A high-level beta wave may be acquired when a human is in a panic state. [9]

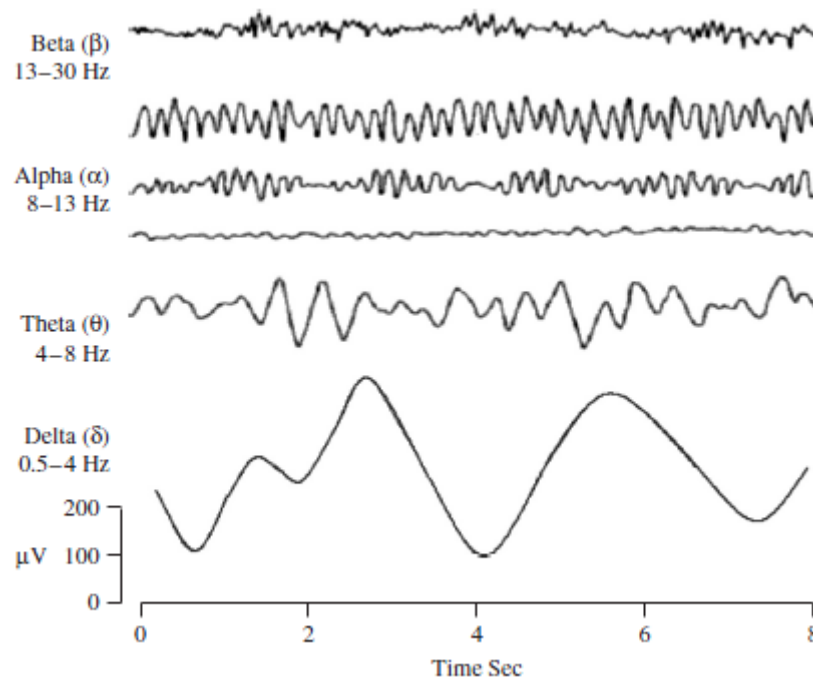


Figure 2: Major brain waves. [2]

## 2.4 Acquiring and recording of an EEG activity

EEG signal is a time variation of potential difference between two electrodes placed on scalp surface of a subject (the invasive method is used in medicine praxis only). EEG signal can be also defined as a weighted summation composed of signals produced by huge amount of neurons placed in parts of brain called cortex and thalamus. Intensity of the neuron groups electric activity depends on distance from the electrode (further neurons contribute less to the EEG than neurons closer to the electrode). There is no way to separate single contribution of one neuron from another. [15]

The commonly used electrodes placement schema is called 10-20 system (the numbers refer to percentage ratio between axis with electrodes), which is shown in Figure 3.

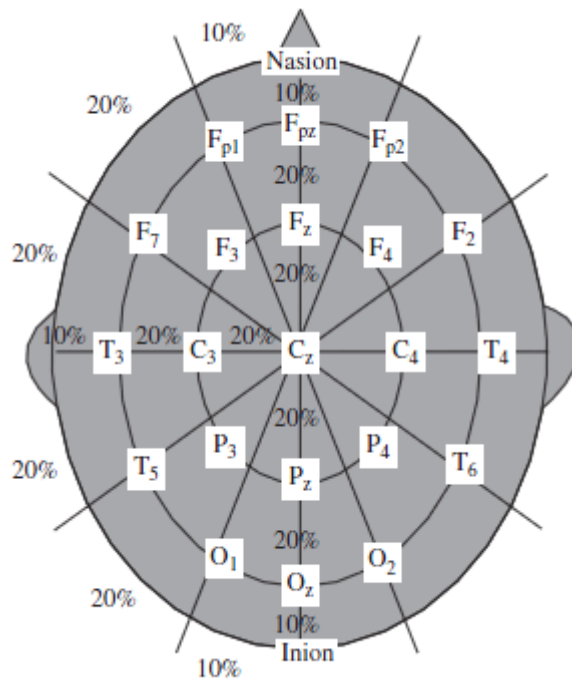


Figure 3: Top view of the 10-20 system for electrodes placement. For a detail view from all perspectives see Appendix B. [2]

The data from electrodes are recorded by an electroencephalograph. Its output is called electroencephalogram. [20]

#### 2.4.1 Electroencephalograph

An electroencephalograph is used for recording of the EEG signal. Because of the low voltage of the EEG activity, its voltage must be amplified by an input amplifier at first. But the goal is to amplify only the EEG activity, not the noise. One way to avoid amplifying the noise is to use the differential amplifier with two inputs:

1. direct input (active)
2. inverted input (reference)

The differential amplifier amplifies the difference in voltage between these two inputs.

#### 2.4.2 Electroencephalogram

Electroencephalogram contains a time variation of potential difference between an electrode and the reference electrode for all used electrodes in a form which is suitable for a next processing. Because the EEG signal is discrete, a linear function is used as an interpolation between each two immediately adjacent samples (see Figure 4).

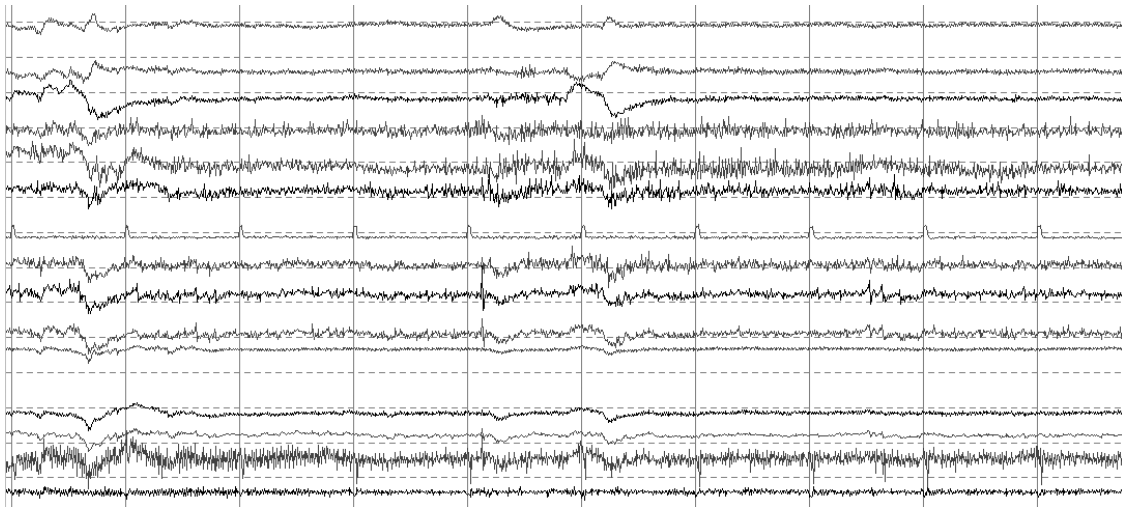


Figure 4: An example of an EEG signal. Ten seconds of the record which was sampled with 1 kHz frequency is shown. [20]

## 2.5 Artifacts

Artifacts are signals with non-cerebral origin which appear in EEG signal. Artifacts are divided into two following categories:

- **Artifacts with biological origin:** All muscles in the human body are controlled by electrical impulses. These impulses spread in CNS from the brain to muscles and produce much more electrical activity than the brain activity monitored during EEG measurement. Biological artifacts are e.g. head movements, swallow, eye movements, perspiration, etc. [20]
- **Artifacts from the surrounding electromagnetic field,** e.g. 50 Hz from mains.

An example of a few artifacts is shown in Figure 5.

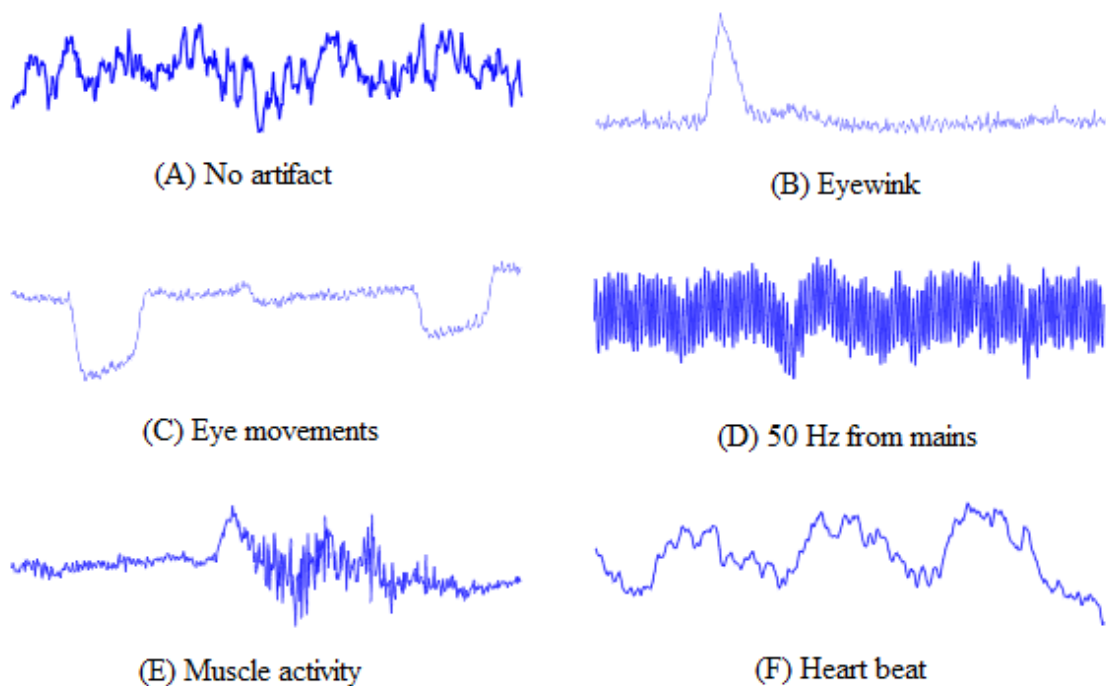


Figure 5: Examples of artifacts. [16]

To detect artifacts, two following criterions are usually used:

1. **Amplitude criterion:** This criterion is based on an empirically verified fact that during the common EEG activity (even if ERPs are present), no higher amplitudes than  $30\mu\text{V}$  occur. When the artifacts are detecting on an averaged data, the amplitude threshold value can be lower. For this criterion, the baseline correction (see Chapter 3.6) is a necessary preprocessing method. [20]
2. **Gradient criterion:** This criterion is based on monitoring of difference between each two immediately adjacent functional values. The artifact is detected for the difference higher than a chosen threshold. By definition, this criterion is not sensitive to a baseline. [20]

### 3 Event-related potentials

An event-related potential (ERP) is the measured brain response that is the direct result of a specific sensory, cognitive, or motor event. More formally, it is any stereotyped electrophysiological response to a stimulus [15]. Experimental neurologists discover many different kinds of stimuli which evoke ERPs.

#### 3.1 ERP waveform

There are three properties which describe an ERP waveform (see Figure 6):

- latency
- frequency
- amplitude

The amplitude represents the rate of neural activity as the response to the stimulus [22]. The time between the stimuli occurrence and the response occurrence is considered as the time necessary for information processing in the brain and it is called latency [20]. The frequency characterizes the ERP waveform.

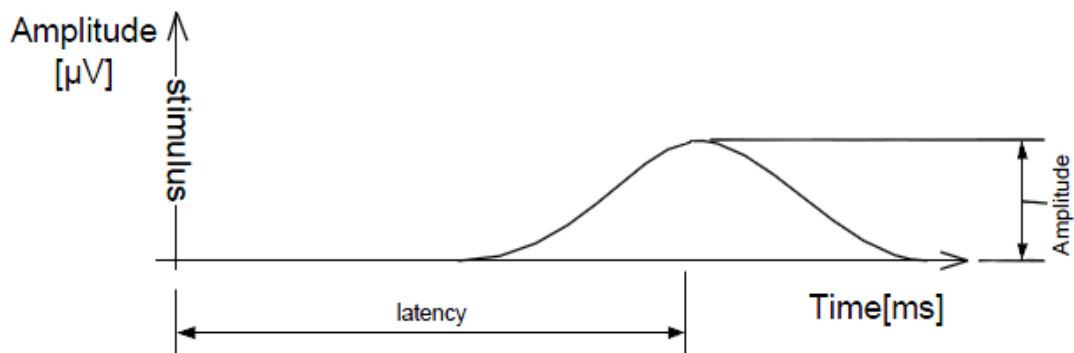


Figure 6: Properties of an ERP waveform. [22]

#### 3.2 ERPs naming conventions

Notwithstanding some ERPs have acronyms; most of them are signed with a string composed from a character and a number or numbers. The character signs polarity of the ERP waveform:

- **P** for positive waveform
- **N** for negative waveform
- **C** for waveforms which have not dedicated one polarity

If the character is followed by a one-digit number, then the number says the order of the wave. For example N4 is the fourth negative ERP waveform. Otherwise, the character is followed by a three-digit number, which express an accurate latency of the ERP waveform in milliseconds. For example P100 is the positive waveform which appears 100 ms after a stimulus.

Generally speaking, there is no equation in the way that N1 = N100, P2 = P200, etc. A good example is the N2 waveform. It is the second negative waveform, but it appears

typically around 400 ms after a stimulus. So it can be signed as N400 or N2, but definitely not as N200.

In Figure 7, a few well-known ERP waveforms are show:

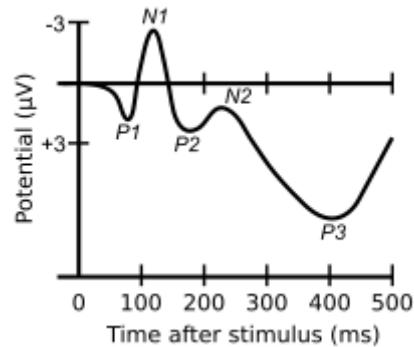


Figure 7: Typical amplitudes, frequencies, latencies, and waveforms of well-known ERPs. [15]

### 3.3 Major ERP components

Note that this is not a complete list of all known ERPs.

#### 3.3.1 Visual sensory responses

- **C1** The first major visual ERP component is usually called the C1 wave. Unlike most other components, it is not labeled with a P or an N because its polarity can vary. The C1 wave typically onsets 40-60 ms poststimulus and peaks 80-100 ms poststimulus, and it is highly sensitive to stimulus parameters, such as contrast and spatial frequency. [15]
- **P1** The C1 wave is followed by the P1 wave, which typically onsets 60-90 ms poststimulus with a peak between 100-130 ms. Note that the P1 onset time is difficult to assess accurately due to overlap with the C1 wave. In addition, P1 latency will vary substantially depending on stimulus contrast. [15]
- **N1** The P1 wave is followed by the N1 wave. There are several visual N1 subcomponents. The earliest subcomponent peaks 100-150 ms poststimulus, and there appear to be at least two posterior N1 components that typically peak 150-200 ms poststimulus. [15]
- **P2** A distinct P2 wave follows the N1 wave. This component is larger for stimuli containing target features. The P2 wave is often difficult to distinguish from the overlapping N1, N2, and P3 waves. [15]

#### 3.3.2 Auditory sensory responses

- **N1** Like the visual N1 wave, the auditory N1 wave has three distinct subcomponents which peak around 75 ms, 100 ms, and 150 ms. The N1 wave is sensitive to attention. [15]

- **Mismatch negativity (MMN)** The MMN is observed when subjects are exposed to a repetitive train of identical stimuli with occasional mismatching stimuli (e.g. a sequence with many 800Hz tones and occasional 1200Hz tones). The mismatching stimuli elicit a negative-going wave and typically peaks between 160 and 220 ms. [15]
- **The N2 family** Clearly different components were identified in the N2 time range. A repetitive, nontarget stimulus will elicit an N2 deflection that can be thought of as the basic N2. If other stimuli are occasionally presented within a repetitive train, larger amplitude is observed in the N2 latency range. If these deviants are task-irrelevant tones, this effect will consist of mismatch negativity. If the deviants are task-relevant, then a somewhat later N2 effect is also observed, called N2b (the mismatch negativity is sometimes called N2a). This component is larger for less frequent targets, and it is thought to be a sign of the stimulus categorization process. Both auditory and visual deviants will, if task-relevant, elicit an N2b component. [15]
- **The P3 family** There are several distinguishable ERP components in the time range of the P3 wave. The two major components are P3a and P3b. Both are elicited by unpredictable, infrequent shifts in tone pitch or intensity, but the P3b component is present only when these shifts are task-relevant. When ERP researchers refer to the P3 component, they almost always mean the P3b component (this also applies for this thesis). [15]

### 3.4 Averaging

The amplitude of ERPs is quite low, up to 30  $\mu\text{V}$  (the background EEG activity has amplitudes even 100  $\mu\text{V}$ ). Therefore, it is necessary to use averaging technique to highlight them and suppress the background EEG [9; 23].

Averaging is the common method for highlighting of ERP waveforms. During the averaging of the same kind of ERP waveforms, the noise is reduced and the waveform is highlighted (see Figure 8).

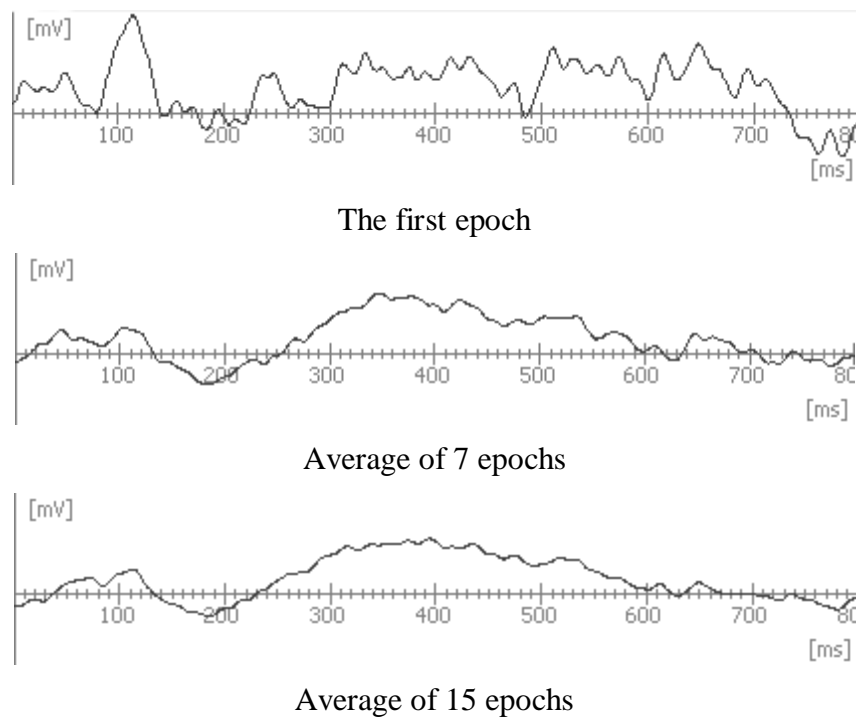


Figure 8: Averaging of epoch which contains the P3 waveform. [20]

On the input of the averaging method, there is a set of epochs. Generally speaking, it is not a good idea to take all target-epochs from an ERP experiment and use them for averaging – some ERP waveforms could be significantly shifted, some target epochs could not contain the ERP waveform at all, some ERP waveforms could be affected by an artifact which cannot be / was not detected by an artifact rejection method.

It seems to be useful to have a method which would determinate whether use the epoch for averaging or not. The analysis of variance seems to be the suitable method.

### 3.5 Analysis of variance for selecting epochs for averaging

#### 3.5.1 Principle

For purposes of the analysis of variance (ANOVA), let the epoch's index be the first variable, and the epoch's functional values be the second variable. If all epochs contain the same ERP waveform, they are similar to each other. It means that there is no dependency between epoch's index and epoch's functional values. If there is an epoch which does not contain the ERP waveform, this epoch is not similar to the other epochs. It means that there is a dependency between epoch's index and epoch's functional values. [20]

The basic principle of ANOVA follows (for a detailed description of ANOVA principle see [24]). There is the following table on the input:



<i>Index of an epoch</i>	<i>Epoch's functional values</i>	<i>Sum of the functional values</i>	<i>Average of the functional values</i>
1	$y_{11}, y_{12}, y_{13}, \dots, y_{1n}$	$y_1$	$\bar{y}_1$
2	$y_{21}, y_{22}, y_{23}, \dots, y_2$	$y_2$	$\bar{y}_2$
...	...	...	...
k	$y_{k1}, y_{k2}, y_{k3}, \dots, y_{kn}$	$y_k$	$\bar{y}_k$
<i>Sum</i>		$y$	$\bar{y}$

Table 1: ANOVA input table.

The values of  $y_i$  and  $\bar{y}_i$  should be approximately equal for  $i \in 1, \dots, k$ . The degree of difference (called determinative ratio) is defined as follows:

Intergroup sum of squares:

$$S_{ym} = \sum_{i=1}^k n_i (\bar{y}_i - \bar{y})^2 \quad (1)$$

Intergroup sum of squares:

$$S_{yv} = \sum_{i=1}^k \sum_{j=1}^{n_i} (y_{ij} - \bar{y}_i)^2 \quad (2)$$

Total sum of squares:

$$S_y = \sum_{i=1}^k \sum_{j=1}^{n_i} (y_{ij} - \bar{y})^2 \quad (3)$$

Following equation is valid:

$$S_y = S_{ym} + S_{yv} \quad (4)$$

The determinative ratio is defined as follows:

$$P^2 = \frac{S_{ym}}{S_y} \in \langle 0, 1 \rangle \quad (5)$$

The closer the value  $P^2$  is to 1, the bigger is the difference between epochs. In [15], the 0.05 is recommended as a threshold value when the epochs are similar enough. It means that there is a 95% probability that the similarity between the epochs is not a coincidence. [20]

### 3.5.2 Practical use

Into the set for epochs averaging, one epoch by one are added. If the determinative ratio exceeds the threshold value, then the epoch is not suitable for averaging and therefore it is rejected from the averaging set. It is easy to see that the first epoch must be suitable for averaging. [20]

### **3.6 Baseline**

When the ERP waveform amplitude is measured, the voltage is usually affected by an average prestimulus voltage. The average prestimulus voltage is called baseline value except of the case when its value is zero. The baseline can have a significant effect on the result of every method for ERP processing (detection algorithms, ERP waveforms averaging method, etc.). Therefore, it is necessary to compensate the base in every ERP epoch. In [15], it is recommended to calculate the average voltage in the 200 ms before stimulus onset and then subtract the baseline value from every single functional value of the epoch.

### **3.7 10 simple rules for designing ERP experiments**

In [21], following 10 rules for designing a successful ERP experiment were published:

1. Peaks and components are not the same thing. There is nothing special about the point at which the voltage reaches a local maximum or minimum.
2. It is impossible to estimate the time course or peak latency of a latent ERP component by looking at a single ERP waveform – there may be no obvious relationship between the shape of a local part of the waveform and the underlying latent components.
3. It is extremely dangerous to compare an experimental effect (i.e., the difference between two ERP waveforms) with the raw ERP waveforms.
4. Differences in peak amplitude do not necessarily correspond to differences in component size, and differences in peak latency do not necessarily correspond to changes in component timing.
5. Never assume that an averaged ERP waveform accurately represents the single-trial waveforms.
6. Whenever possible, avoid physical stimulus confounds by using the same physical stimuli across different psychological conditions. This includes “context” confounds, such as differences in sequential order.
7. When physical stimulus confounds are unavoidable, conduct control experiments to assess their plausibility. Never assume that a small physical stimulus difference cannot explain an ERP effect (even at a long latency).
8. Be cautious when comparing averaged ERPs that are based on different numbers of trials.
9. Be cautious when the presence or timing of motor responses differs between conditions.
10. Whenever possible, vary experimental conditions within trial blocks rather than between trial blocks.

## 4 Brain-computer interface

Brain-computer interfacing (BCI) (also called brain-machine interfacing (BMI)) is a challenging problem that forms part of a wider area of research, namely human-computer interfacing (HCI), which interlinks thought to action. BCI can potentially provide a link between the brain and the physical world without any physical contact. In BCI systems the user messages or commands do not depend on the normal output channels of the brain [8]. Therefore the main objectives of BCI are to manipulate the electrical signals generated by the neurons of the brain and generate the necessary signals to control some external systems [9].

To avoiding of misunderstanding, a few definitions from journal articles and technical papers follow:

- A BCI, sometimes called a direct neural interface or a brain-machine interface, is a direct communication pathway between a human or animal brain (or brain cell culture) and an external device. In one-way BCIs, computers either accept commands from the brain or send signals to it (for example, to restore vision) but not both. [3]
- A BCI is a communication system in which messages or commands that an individual sends to the external world do not pass through the brain's normal output pathways of peripheral nerves and muscles [1].
- BCI systems measure specific features of brain activity and translate them into device control signals [2].

From a practical point of view, an EEG/ERP based BCI system can be used to control external devices such as computers, wheelchairs or virtual environments. One of the most important applications is a spelling device to aid severely disabled individuals with communication, for example people disabled by amyotrophic lateral sclerosis. [14]

From BCI definitions above, the fact that there is no exact definition of a brain activity which has to be used for a BCI is resulting. A BCI can be based on non-invasive methods of monitoring brain activity, such as magnetoencephalography (MEG), positron emission tomography (PET) and functional magnetic resonance imaging (fMRI) (see [5; 6; 7]) [4]. However, these methods for brain activity monitoring are not portable, technically demanding, and very expensive (see Appendix A for comparison). Therefore, they are not convenient for use in BCI systems.

Looking at the reasons why MEG, PET, and fMRI are not convenient for use in BCI systems, it is clear that EEG is not affected by any of them. An electroencephalograph is a portable, relatively not technically demanding, and – in comparison with MEG, PET, and fMRI – cheap method for monitoring of the brain activity.

Basically, there are three different principles a BCI system in EEG domain can be based on:

- changes in EEG activity
- event-related potentials
- steady state visually evoked potentials (SSVEP)

In following chapters, the emphasis will be placed on the principles of BCIs based on ERPs and SSVEPs. BCI systems based on changes in EEG activity are out of scope of this thesis.

#### 4.1 The parts of an ERP based BCI system

A schema of the BCI based on ERP is shown in Figure 10. Its principle is – in a nutshell – following: The user who controls the BCI is stimulated by a stimulator. The stimulation is performed in such a way, that the result of the stimulation is an ERP (most often the P3) synchronized in time that it is possible to determine which action should be invoked in the controlled application. The ERP is captured by electroencephalograph along with other brain activity (EEG) and with signal with non-cerebral origin (artifacts, etc.). Then, the signal is preprocessed by methods into a form acceptable by the feature extraction process. When features are extracted, the classifier decides whether the signal contains the ERP or not. If yes, the desired action is identified by the extraction of meaning module (using synchronization with the stimulator) and is sent to the controlled application. Otherwise, the stimulator produces a new stimulus.

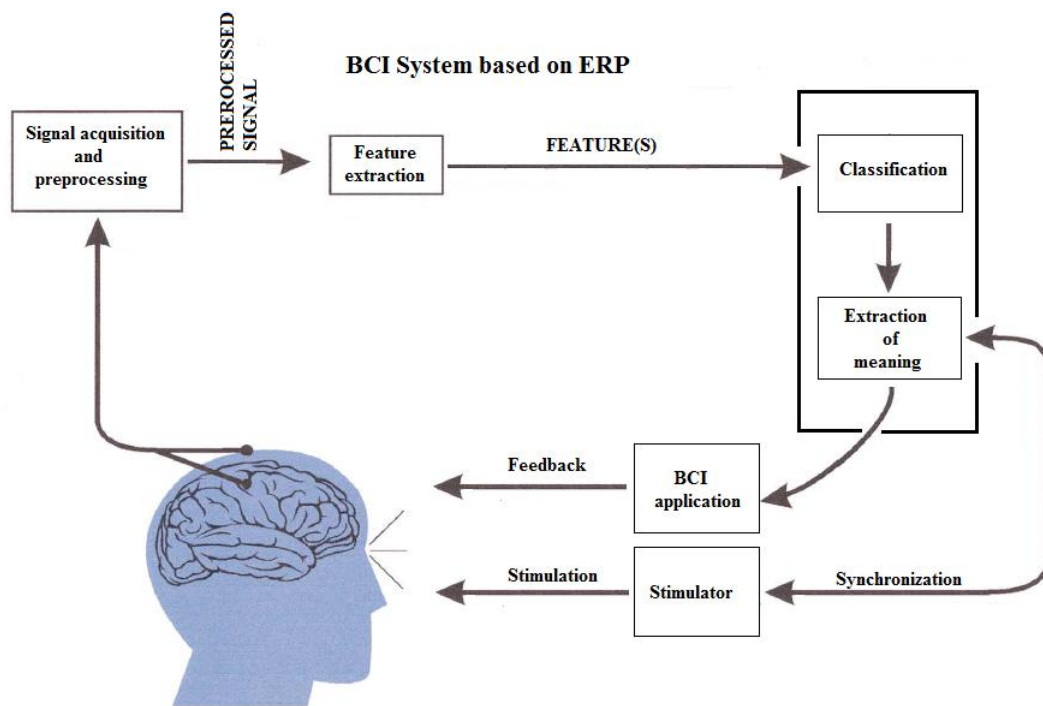


Figure 10: Generall BCI schema.

Of course, the process described above is too general to cover all on ERPs based BCI systems. Following chapters describes modules of the BCI system one by one considering P3 based BCI (example of a BCI calculator) and SSVEP based BCI (example of a BCI speller).

### 4.1.1 Stimulator

The stimulator is a very important part of every BCI. From BCI user's point of view, stimuli produced by the stimulator create the user interface (UI).

#### 4.1.1.1 P3 based

The stimulation is based on odd-ball paradigm – two easy identifiable stimuli are delivered repeatedly. One stimulus is delivered with significantly lower probability than the second one. The stimulus with lower probability of delivering is called target stimulus, the stimulus with higher probability of delivering is called non-target stimulus. The difference between these two stimuli is that the target stimulus is followed by the P3 waveform in contrast to the non-target stimulus which is not followed by the P3 waveform.

The UI for BCI calculator is shown in Figure 11. The whole rows and whole columns are flashing. The flashing is happening in series. At each series, all rows and all columns flash once in a random order. The next series is not started until all rows and all columns flashed. The number of series depends on used preprocessing method and BCI design.

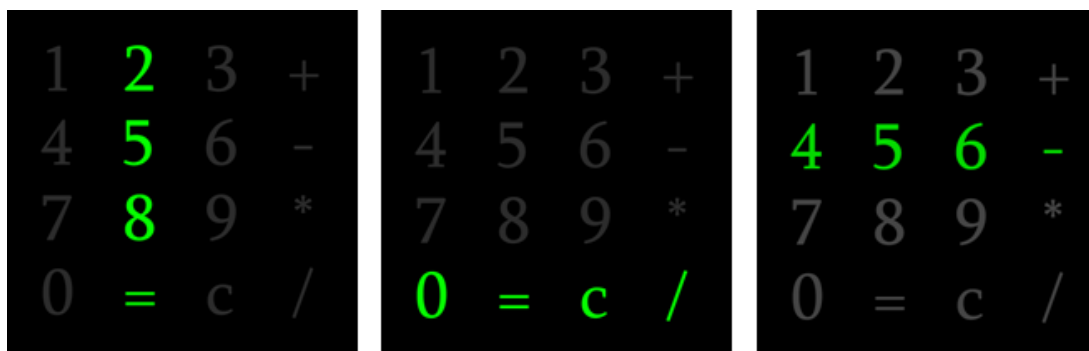


Figure 11: BCI calculator UI.

The user focuses on the number or operation which he wants to enter. Its flash evokes the P3 potential. Information about the time of each row/column flash is stored and synchronized with the acquired EEG/ERP signal. This synchronization is necessary for the extraction of meaning module.

In study [14], it was proved that for this kind of BCI, it is more suitable use flashing rows and columns instead of flashing of single characters (from detection accuracy point of view).

#### 4.1.1.2 SSVEP based

In a BCI system based on steady-state visual evoked potentials (SSVEP), the system reflects the user's attention to an oscillating visual stimulus [12]. Therefore, flashing lights, flashing panels or flashing UI elements are usually used. Their responses appear in the visual cortex and correspond to SSVEPs at the same frequencies and higher harmonics [13].

The UI for BCI spelling system is shown in Figure 12. Arrows and the Select button (control elements) are flashing – each of them with a different frequency which cannot

be masked by the other frequencies (including harmonic frequencies) or the basic EEG rhythms.

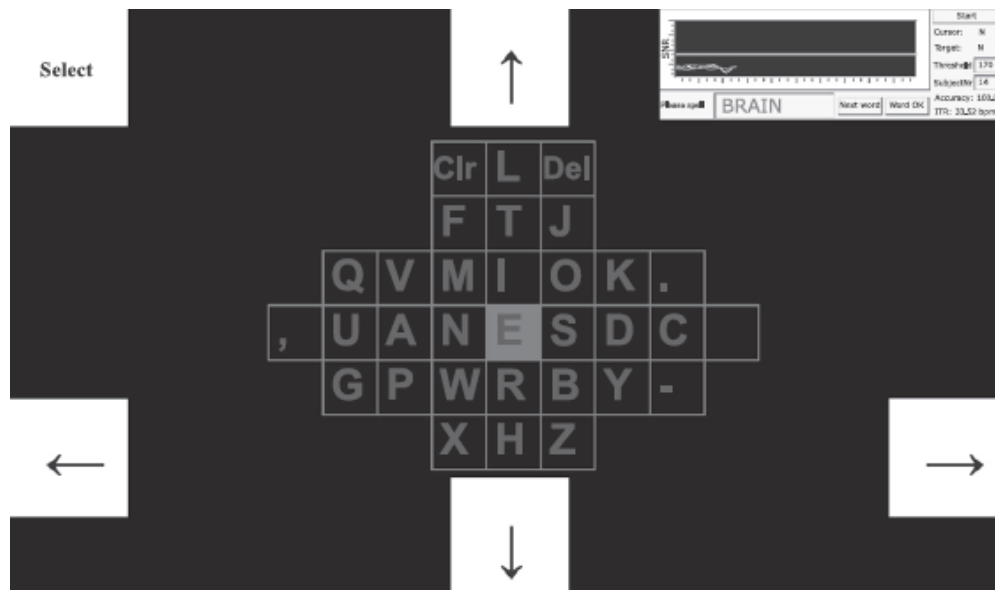


Figure 12: Example of UI for BCI spelling system based on SSVEP. [11]

The user focuses on one of control elements to move the cursor over the desired character or to confirm selection of the desired character. BCIs based on SSVEP do not need time synchronization between flashing elements and EEG/ERP signal.

#### 4.1.2 Signal acquisition

The raw EEG/ERP signals have amplitudes of the order of  $\mu$ volts and contain frequency components of up to 300 Hz. To retain the effective information the signals have to be amplified before the ADC and filtered, either before or after the ADC, to reduce the noise and make the signals suitable for processing and visualization. The filters are designed in such a way not to introduce any change or distortion to the signals. Highpass filters with a cut-off frequency of usually less than 0.5 Hz are used to remove the disturbing very low frequency components such as those of breathing. On the other hand, high-frequency noise is mitigated by using lowpass filters with a cut-off frequency of approximately 50–70 Hz. Notch filters with a null frequency of 50 Hz are often necessary to ensure perfect rejection of the strong 50 Hz power supply. In this case the sampling frequency can be as low as twice the bandwidth commonly used by most EEG systems. The commonly used sampling frequencies for EEG recordings are 100, 250, 500, 1000, and 2000 samples/s. [9]

Usually, Fz, Cz, Pz, O1, and O2 electrodes are used.

#### 4.1.3 Signal preprocessing

##### 4.1.3.1 P3 based

The signal preprocessing methods for BCI based on P3 are as follows:

1. The EEG/ERP signal is divided into epochs. Each epoch starts with the time when a stimulus appeared and is so long that the whole P3 waveform fits into it (usually around 500 ms).

In the Figure 14, the first minute of stimulation is shown. The epochs are highlighted by green color in their whole length.

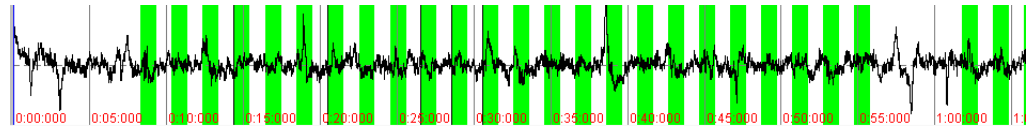


Figure 14: The time without stimulation is visible at start. Then the stimulation is started. Three series of flashing are performed (24 epochs). It is enough to detect one number or operation. Then, a short pause can be seen and then, entering of the next number or operation is started.

2. Epochs which contain artifacts are rejected from the next processing.
3. Baseline is corrected in all epochs.
4. Epochs related with the same stimulus are averaged and only the averages go to the next processing.

#### **4.1.3.2 SSVEP based**

Because the undesirable frequencies are filtered on signal acquisition level, the only useful preprocessing method for SSVEP based BCI is the artifacts rejection (no dividing into epochs or averaging is necessary and baseline correction is unnecessary because it does not have any impact on SSVEP frequencies).

#### 4.1.4 Feature extraction

This phase is closely related to the used classification method:

##### **4.1.4.1 P3 based**

- For artificial neural networks is this phase realized by a feature vector extraction method.
- For the detection based on the matching pursuit algorithm is this phase realized by the matching pursuit algorithm itself. The same is valid for continuous wavelet transform and discrete wavelet transform.
- For Hilbert-Huang transform is this phase realized by empirical mode decomposition.

##### **4.1.4.2 SSVEP based**

- Usually, the discrete short time Fourier transform is used.

#### 4.1.5 Classification

##### **4.1.5.1 P3 based**

Following methods suitable for P3 detection are described in this thesis:

- linear discriminant analysis

- continuous wavelet transform
- discrete wavelet transform
- matching pursuit algorithm
- Hilbert-Huang transform
- self-organizing map
- ART2 neural network

Of course, this is not a complete list of algorithms and methods which are suitable for P3 waveform detection, but these methods cover all mainstream approaches.

#### **4.1.5.2 SSVEP based**

Fast Fourier transform is a suitable method for SSVEP detection and is described in this thesis. Usually the frequencies are classified into  $N+1$  classes where  $N$  is the number of frequencies used for stimulation. The  $N+1$ th class represents the *do nothing* state.

#### 4.1.6 Extraction of meaning

##### **4.1.6.1 P3 based**

Decision whether the epoch contains the P3 waveform or not along with related stimuli are on the input of the extraction of meaning module. When the decision is positive (the epoch contains the P3 waveform), the action assigned to the related stimuli is performed (in case of the ERP calculator, a number or an operand is entered).

In the Figure 15, the result of the mathematical example  $2 + 4 =$  is shown. Epochs of each row and column for the number 2 are viewed. The P3 waveforms are visible in column 2 and row 1. Therefore the number 2 was detected (it is its position in the matrix of numbers and operands – see Figure 11).



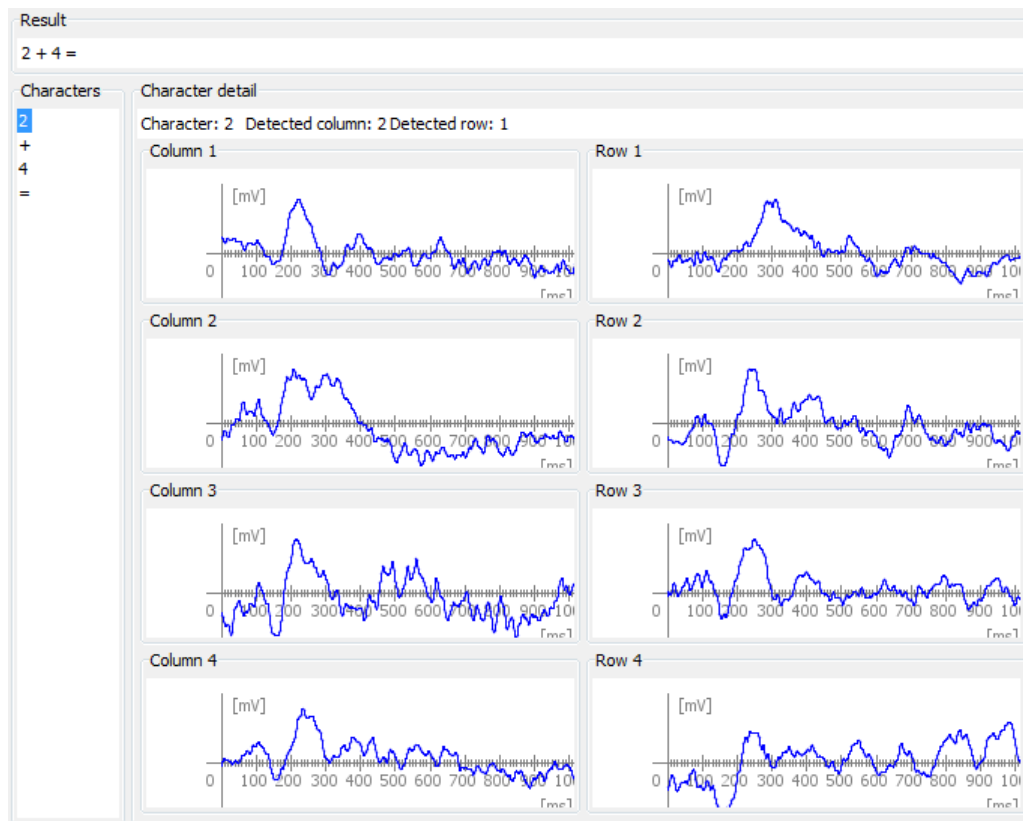


Figure 15: Epochs of each row and column related to number two.

#### 4.1.6.2 SSVEP based

Extraction of meaning in case of SSVEP based BCI is quite simple. The related action is performed when one of the frequencies which are used for stimulation is detected.

### 4.2 Is everybody able to use an ERP based BCI system?

Looking at the BCI principle a question arises: Is everybody able to use an ERP based BCI system? Of course, from neurological point of view, the ERPs are observable for all healthy people. But the question is whether they can use them for controlling a system via BCI. An interesting study about this issue is [14]:

Two BCI based on P3 for spelling were used for characters entering accuracy testing. The first one was used flashing rows and columns, the second one used single character flashing. Both BCI used the same UI – a matrix with 36 characters. During the experiment, one hundred subjects used the BCIs to spell the word WATER in five minutes. Data acquired during this phase were used as a learning data for the detection algorithm. Then, the subjects spelled the word LUCAS and the detection accuracy was calculated for each letter separately. When the spelling was done, subjects filled out a questionnaire about age, sex, education, sleep duration, working duration, cigarette consumption, and coffee consumption.

Results are following: 72.8 % of the subjects were able to spell with 100% accuracy with the rows/column flashing and 55.3 % of the subjects spelled with 100% accuracy

in the single characters flashing. Less than 3% of the subjects did not spell any character correctly. People who slept less than 8 h performed significantly better than other subjects. Sex, education, working duration, and cigarette and coffee consumption were not statistically related to differences in detection accuracy. This study shows that high spelling accuracy can be achieved with the P300 BCI system using approximately 5 min of training data for a large number of non-disabled subjects, and that the rows/column flashing is superior to the single character flashing. 89 % subjects were able to spell with accuracy 80–100 % with rows/column flashing. [14]

## 5 Methods suitable for ERP waveforms detection

### 5.1 Linear discriminant analysis

Linear discriminant analysis (LDA, known also as Fisher's linear discriminant, after its inventor) is a commonly used technique for data classification and dimensionality reduction. LDA easily handles the case where the within-class frequencies are unequal. This method maximizes the ratio of between-class variance to the within-class variance in any particular data set thereby guaranteeing maximal separability. [39]

For demonstration of LDA principle, let the data are displayable in a two dimensional system, and let the data are separable into  $C$  classes:

$$set\ 1 = \begin{bmatrix} s1_{11} & s1_{12} \\ \dots & \dots \\ s1_{m1} & s1_{m2} \end{bmatrix}, set\ 2 = \begin{bmatrix} s2_{11} & s2_{12} \\ \dots & \dots \\ s2_{n1} & s2_{n2} \end{bmatrix}, \dots, set\ C = \begin{bmatrix} sC_{11} & sC_{12} \\ \dots & \dots \\ sC_{o1} & sC_{o2} \end{bmatrix} \quad (6)$$

Then, operations related with LDA are as follows:

- Let  $\mu_i$  be the mean vector of set  $i$ ,  $i = 1, 2, \dots, C$
- Let  $M_i$  be the number of samples within set  $i$ ,  $i = 1, 2, \dots, C$
- Let  $M = \sum_{i=1}^C M_i$  be the total number of samples

Then,

- the within-class scatter matrix is defined as follows:

$$S_w = \sum_{i=1}^C \sum_{j=1}^{M_i} (y_j - \mu_i)(y_j - \mu_i)^T \quad (7)$$

- the between-class scatter matrix is defined as follows:

$$S_b = \sum_{i=1}^C (\mu_i - \mu)(\mu_i - \mu)^T \quad (8)$$

where  $\mu = \frac{1}{C} \sum_{i=1}^C \mu_i$  is mean of the entire dataset

LDA computes a transformation (using eigen vectors) that maximizes the between-class scatter while minimizing the within-class scatter (for detailed description of the maximization principle, see [39]):

$$\text{maximize } \frac{\det(S_b)}{\det(S_w)} \quad (9)$$

The example of two-set problem is shown in Figure 16.

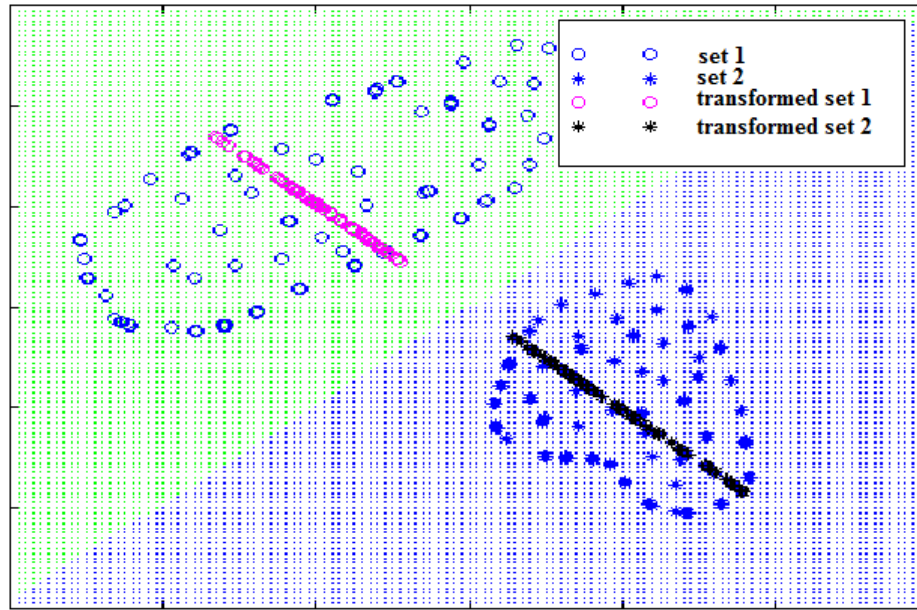


Figure 16: LDA for two-class problem. [39]

### 5.1.1 ERP detection with LDA

In general,  $N$ -dimensional feature vector is obtained from each epoch. Then the feature vectors are manually divided into two sets (contain ERP waveform vs. do not contain ERP waveform). The LDA is computed and a linear function (hyperplane) which divides the  $N$ -dimensional space to two subspaces – each one for one set – is established. One subspace contains feature vectors of epochs which contain an ERP waveform. The other subspace contains all other feature vectors.

Let  $f(p_1, p_2, \dots, p_N) = 0$  be the hyperplane. During the classification process, for each feature vector of each epoch, the items are substituted into the hyperplane equation and enumerated. When the result is higher than zero, the feature vector belongs to the first set. Otherwise, the feature vector belongs to the second set.

## 5.2 Fourier transform

The Fourier transform (FT) is an operation that transforms data from the time domain into the frequency domain [25]. According to different fields of its application, FT exists in a few different variants. Note, that Fourier transform expect a periodical signal on its input. For non-periodical signals, this request is usually solved in such a way, that the signal is considered as a one period of a periodical signal.

### 5.2.1 Continuous Fourier transform

The continuous Fourier transform (CFT) is defined as follows:

$$F(f) = \int_{-\infty}^{+\infty} x(t) \cdot e^{-j2\pi ft} dt \quad (10)$$

where the independent variable  $t$  is a time, and  $f$  is a frequency. Then,  $F(f)$  is a characteristic of the frequency in the whole length of the input signal  $x$ .

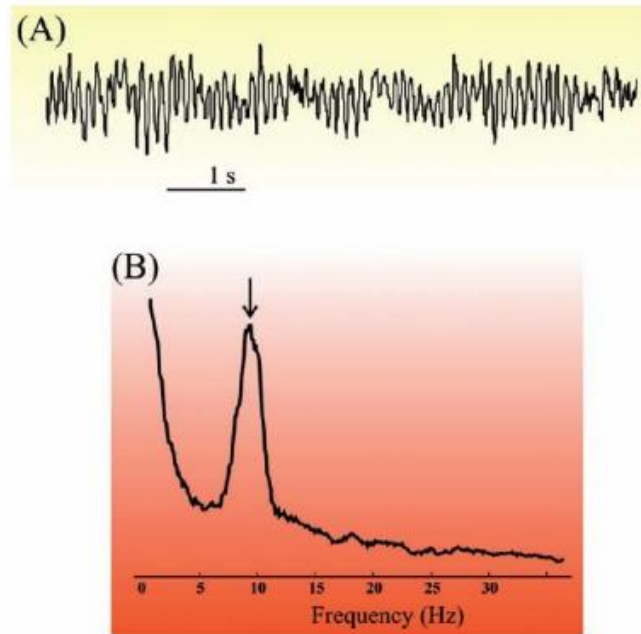


Figure 17: An example of spectral analysis of an EEG trace shown in (A). The trace includes strong oscillation in the alpha band. Accordingly, the power spectrum in (B) shows the clear presence of a component slightly below 10 Hz (arrow) representing this alpha rhythm. For clarity, the spectrum in (B) was smoothed using rectangular 1.5 Hz window. [25]

From both equation (10) and Figure 17 it is clear that the time information is lost during the Fourier transform. This is a disadvantage which makes CFT unusable for ERPs detection as well as for SSVEP detection.

### 5.2.2 Discrete Fourier transform

For use in discrete time typical for computers, the discrete Fourier transform (DFT) is available. The DFT can be computed in three different ways:

- by finding a solution of a system of equations
- by a correlation
- by the fast Fourier transform (FFT) algorithm

Explanation of these methods (especially FFT) is out of scope of this thesis. Principles of all three methods are described in [27].

### 5.2.3 Short-time Fourier transform

The purpose of the short-time Fourier transform (STFT) is to keep a time information when the Fourier transform is performed. The basic idea is to split the input signal by a window. The window is a function which is nonzero on a short interval only (this approach is used in the continuous STFT (CSTFT), in the discrete STFT (DSTFT), the position of the nonzero part of the window is solved by a floating index of the input signal).

Calculating the Fourier transform for windows which covers whole length of the input signal, the Fourier transform of the whole input signal is obtained.

The STFT can be expressed as follows:

$$STFT\{x(t)\} \equiv F(\tau, f) = \int_{-\infty}^{+\infty} x(t)w(t - \tau)e^{-j\omega t} dt \quad (11)$$

where  $w(t)$  is a window,  $x(t)$  is an input signal in the time domain,  $\tau$  is a time shift of the window  $w(t)$ , and  $f$  is the analyzed frequency. In Figure 18, the DSTFT principle is shown.

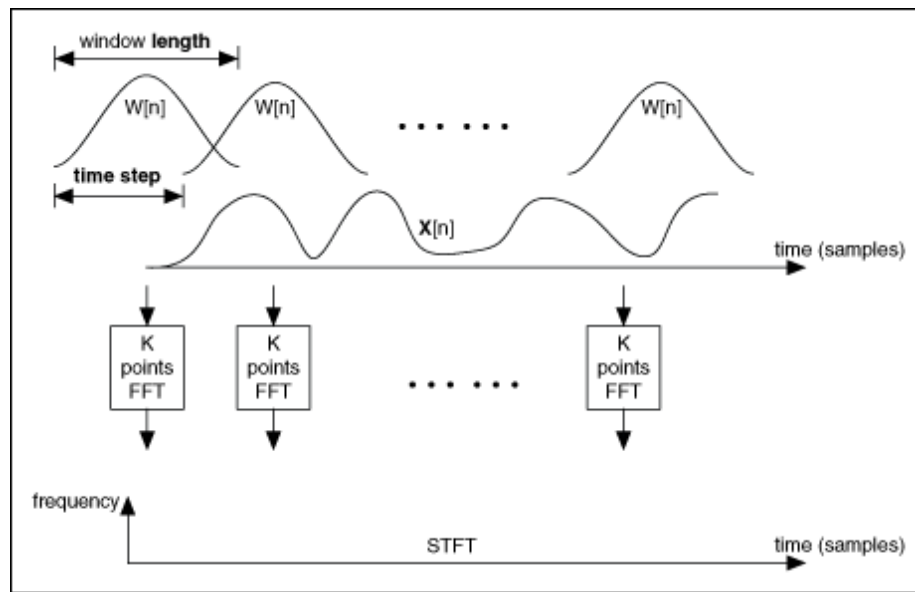


Figure 18: DSTFT principle: The window is shifted over the input signal and the FFT is computed for each window. At the end, we know which frequencies appeared in concrete time intervals. Note that the window positions are a bit overlapping – it is because the fast frequencies which could appear on the border between the windows.[26]

### 5.2.3.1 SSVEP detection with DSTFT

The basic idea is load the EEG/ERP data from the electroencephalograph until the length of the data does not fit the window's length. Then the FFT will be performed. In case that in the FFT result one of the frequencies used for stimulation appear, the SSVEP is detected.

## 5.3 Wavelet transform

Wavelet transform (WT) is a suitable method for analyzing and processing non-stationary signals such as EEG/ERP signal. WT has a good time and frequency localization, which is necessary for ERP detection. For EEG/ERP signal processing it is possible to use Continuous Wavelet Transform (CWT) or Discrete Wavelet Transform (DWT). The wavelet transform is used to divide a time function into called wavelets. See Figure 19 for an example of some well-known wavelet functions.

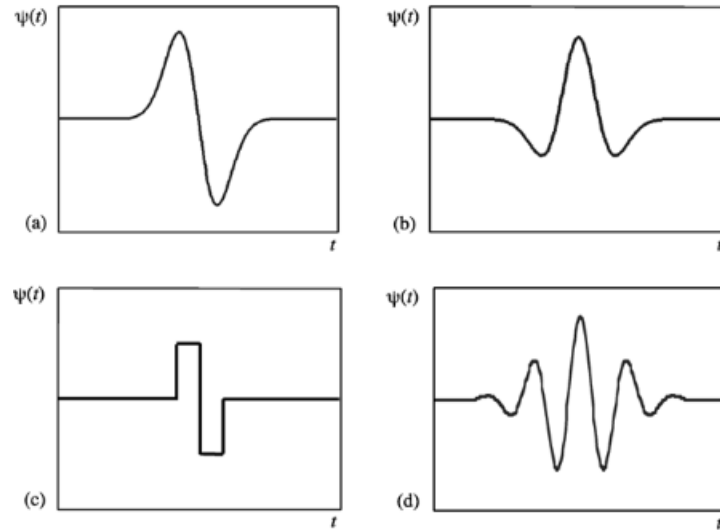


Figure 19: Some well-known wavelets: (a) Gaussian wave, (b) Mexican hat, (c) Haar wavelet, (d) Morlet. [WT6]

### 5.3.1 Principles of continuous wavelet transform

Let  $\psi(t)$  is a wavelet. Then, its functional values for every dilatation  $a$  and translation  $b$  can be expressed as follows:

$$\psi_{u,s}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right) \quad (12)$$

The continuous wavelet transform of a signal  $f$  for the dilatation  $a$  and the translation  $b$  of the wavelet  $\psi$  is defined in [64] as follows:

$$WT(f, u, s) = \int_{-\infty}^{+\infty} f(t) \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right) dt \quad (13)$$

Let's show the principle of CWT with Mexican hat wavelet. Mexican hat is defined as follows:

$$\Psi\left(\frac{t-b}{a}\right) = \left[1 - \frac{(t-b)^2}{a}\right] \cdot e^{-\frac{1}{2}\left[\frac{t-b}{a}\right]^2} \quad (14)$$

where  $a$  (dilatation) corresponds to a frequency, and  $b$  (translation) describes shifting the wavelet over the signal (see Figures 20 and 21). The value of translation is 1, when the CWT is performed.

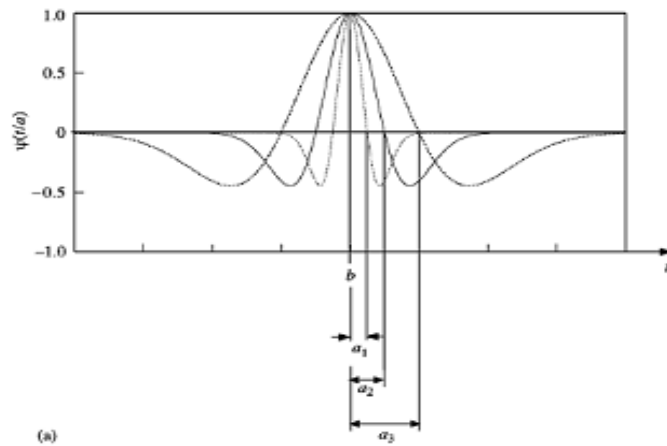


Figure 20: Dilatation of the wavelet (Mexican hat). [37]

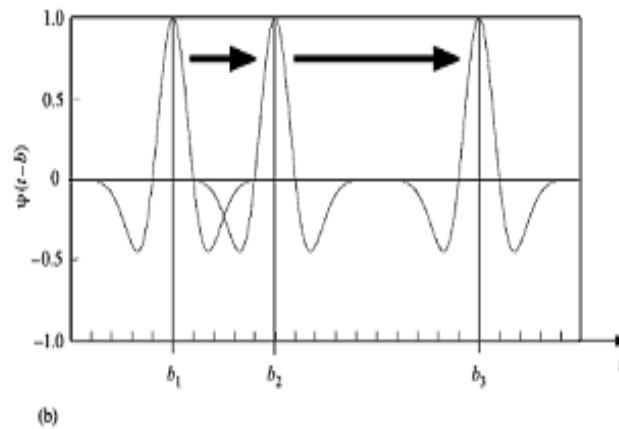


Figure 21: Translation of the wavelet (Mexican hat). [37]

The CWT algorithm can be divided to four following steps:

1. A mother wavelet, starting and ending value of dilatation, step of dilatation, and translation are set.
2. Sum of values of correlation for current dilatation (see Figure 22) and for every translation step to cover the whole signal is computed.
3. The value of dilatation is increased by dilatation step. The algorithm continues with step 2.
4. The calculation is stopped when maximum value of dilatation is reached.



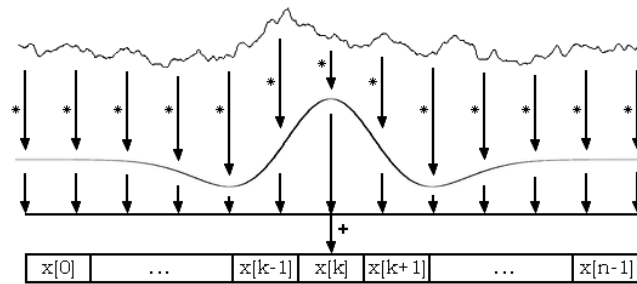


Figure 22: Principle of discrete correlation. [60]

The result of the wavelet transform is visualized in a scalogram, where each coefficient represents a degree of correlation between the transformed wavelet and the signal. Scalogram is gray scaled and the highest values are white (see Figure 23).

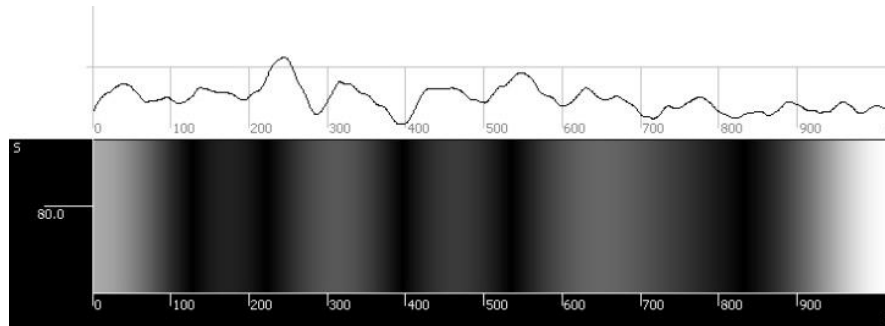


Figure 23: Input signal and its scalogram. [60]

### 5.3.2 Principles of discrete wavelet transform

The continuous wavelet function known from CWT is replaced by two discrete signals – wavelet function and scaling function. See Figure 24 for the Haar wavelet example.

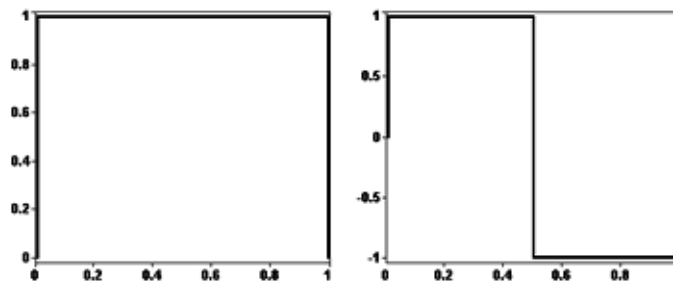


Figure 24: Haar wavelet (scaling function on the left, wavelet function on the right).[60]

Given the limited spectrum band of wavelet functions, the convolution process with this function can be interpreted as a limited band-pass filter [61]. In terms of digital signal processing, wavelet transform can be considered as a bank of filters with signal decomposition into sub-frequency bands. The slowest fundamental frequency components are detected using a scale function. Wavelet function is then documented by a high pass filter and the scale function is a complementary low pass filter. Relevant coefficients are determined taking the convolution of signal and the corresponding analyzing function [60; 62]. The scale is inversely proportional to the frequency; the

low frequencies correspond to large scales and to the dilated wavelet function. Using the wavelet analysis at large scales, we obtain global information from the signal (an approximate component). At small scales we obtain detailed information (a detailed component) representing rapid changes in the signal [62].

Calculation of DWT coefficients is implemented by a gradual application of wavelet function (high frequency filter) and scale function (low frequency filter) to the given signal using Mallatov decomposer scheme (see Figure 25). For each level of decomposition  $p$  so-called detailed component  $D_p(n)$  of the input signal is the output of high pass filter  $h_d(k)$ . The approximation component  $A_p(n)$  is the output of low frequency filter  $l_d(k)$ . Using the convolution and the subsequent subsampling the following equations are valid [62]:

$$D_p(n) = \sum_{k=0}^{L-1} h_d(k)A_{p-1}(2n - k) \quad (15)$$

$$A_p(n) = \sum_{k=0}^{L-1} l_d(k)A_{p-1}(2n - k) \quad (16)$$

for  $n = 0, \dots, N/2$ , where  $A_0(n) = x(n)$  is the analyzed signal, and both sequences  $h_d(k)$  and  $l_d(k)$  define decomposition filters.

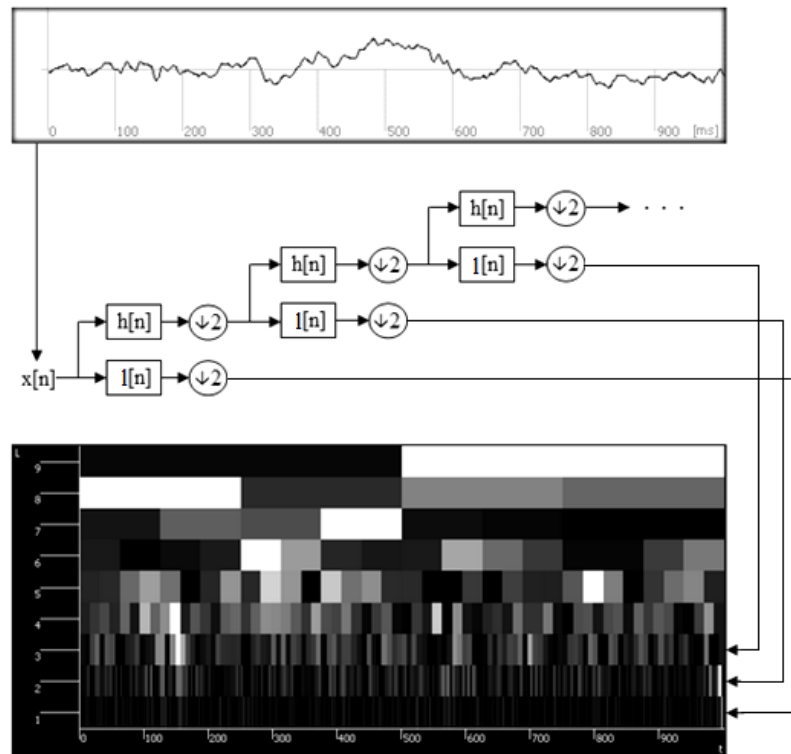


Figure 25: Principle of discrete wavelet transform [60]

### 5.3.3 ERPs detection with WT

When we look for the ERP waveform we compute correlation between a wavelet (which is scaled to correspond to the ERP waveform) and the EEG/ERP signal in the corresponding part of the signal, where the ERP waveform could be situated. This approach avoids a false ERP waveform detection in the signal parts which couldn't contain the ERP waveform. Wavelet coefficients are affected by the match of scaled wavelet and the signal and also by the signal amplitude. When the degree of the correlation is higher than an established threshold, the ERP waveform is considered to be detected. [36]

Of course, the more is the used wavelet similar with the detected ERP waveform, the higher is degree of correlation in case that input signal contains the ERP waveform. The simplest idea is to create a model of the detected ERP waveform and use this model as a wavelet. Unfortunately, there are conditions which have to be valid and which are too strict to enable this approach:

1. The energy of the wavelet must be finite:

$$E = \int_{-\infty}^{+\infty} \Psi(t)^2 dt < \infty \quad (17)$$

where  $E$  is the energy, and  $\Psi$  is the wavelet.

2. If  $\Psi'(f)$  is a Fourier transform of  $\Psi(t)$ , i.e.

$$\Psi'(f) = \int_{-\infty}^{+\infty} \Psi(t) \cdot e^{-i2\pi ft} dt \quad (18)$$

then following condition must be valid:

$$\int_0^{+\infty} \frac{\Psi'(f)^2}{f} df < \infty \quad (19)$$

According to this condition, the average of all functional values of the function  $\Psi(t)$  must be equal to zero.

Because of these conditions and the fact that ERP waveforms do not have average value of all functional values equal to zero, it is necessary to do a lot of tests and choose suitable wavelet empirically as well as the correlation threshold value for ERP waveforms detection.

Following examples show P3 detection with both CWT and DWT. In Figure 26 and Figure 27 the P3 component peak is 480 ms after stimulus; this shift is caused by averaging several epochs. The averaging method is used to improve signal-to-noise ratio (SNR). With higher SNR it is easier to detect ERP. [36]

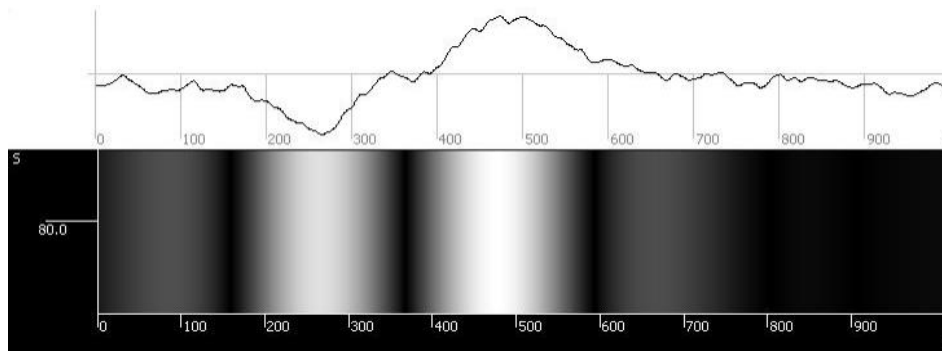


Figure 26: P3 component can be clearly seen in the scalogram (CWT, Mexican hat). [60]

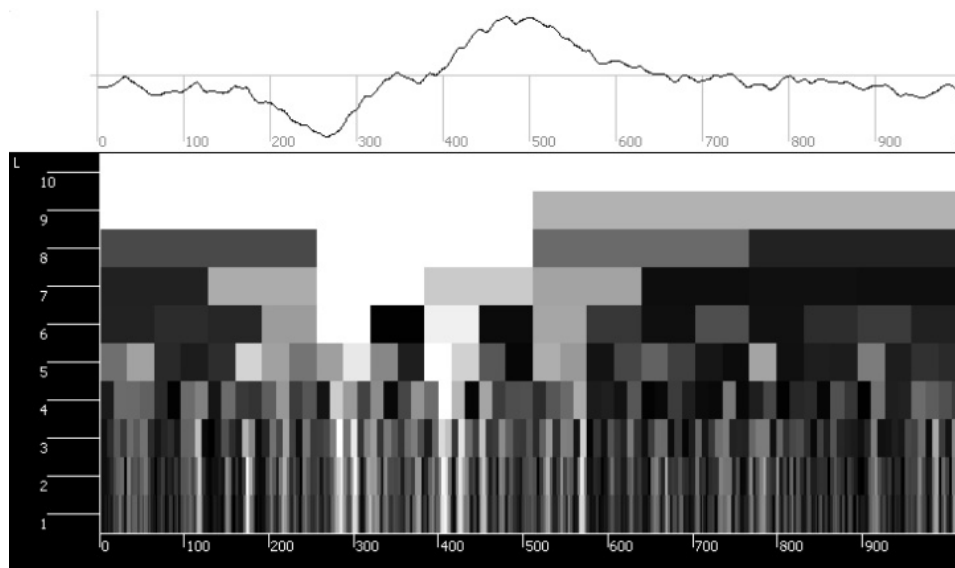


Figure 27: P3 component can be seen in the scalogram (DWT, Haar). [60]

The disadvantage of CWT is its computational complexity, which is linearly growing according to the number of signal samples. Increase of number of input signal samples does not have so heavy impact in case of DWT. To detect ERP components, we tested 1 kHz sampling frequency. Then the epoch, which has to be at least one second long, has 1024 samples. CWT computation on 1024 samples takes approximately 0.75 second. Therefore CWT is not suitable for BCI application in contrast to DWT. We can say that the time of DWT computation on the same sample is insignificant. [36]

## 5.4 Matching pursuit algorithm

### 5.4.1 Basic principle

The matching pursuit (MP) algorithm decomposes any signal to the sum of so-called atoms, which are selected from the dictionary. That means that the input signal  $x$  can be express by atoms  $g_n$  and suitable constants  $a_n$  as follows:

$$x \approx \sum_{n=0}^{N-1} a_n g_n \quad (20)$$

The atom that most closely approximates the input signal is chosen during each iteration. This atom is subtracted from the input signal and the residue enters the next iteration of the algorithm. The total sum of atoms selected successively in algorithm iterations is an approximation of the original signal – more iterations we do, more accurate approximation we get [43]. The error difference between input signal and its approximation leads to zero with growing number of iterations of the MP algorithm.

The MP algorithm is most often associated with Gabor atoms dictionary. Gabor atoms are defined as the Gaussian window:

$$g(t) = e^{-\pi t^2} \quad (21)$$

modulated using cosine function as follows:

$$g_{(s,u,v,w)}(t) = g\left(\frac{t-u}{s}\right) \cdot \cos(vt+w) \quad (22)$$

Each atom is uniquely defined by a ordered quadruple  $(s,u,v,w)$  where  $s$  means scale,  $u$  is shift,  $v$  is frequency, and  $w$  is phase shift.

Let  $\gamma = (s,u,v,w)$  is a modulation vector, let  $\gamma_k = (s,u,v,w)$  is a modulation vector chosen in  $k^{\text{th}}$  iteration, and let  $f$  is an input signal. The criterion for choosing  $\gamma_k$  at each iteration is a scalar product  $\langle f, \gamma \rangle$  which is maximal for  $\gamma_k$ . With respect to this, it is possible to express the input signal after one iteration as follows:

$$f = \langle f, g_{\gamma_0} \rangle \cdot g_{\gamma_0} + Rf \quad (23)$$

where  $Rf$  is a difference between  $f$  and  $g_{\gamma_0}$ . Following formula is a generalization for  $M$  iterations of MP algorithm:

$$f = \sum_{i=0}^{M-1} \langle R^i \cdot f, g_{\gamma_i} \rangle \cdot g_{\gamma_i} + R^M \cdot f \quad (24)$$

where  $R^M \cdot f$  is a difference between  $f$  and sum of all  $M$  Gabor atoms. Figure 28 shows three iterations of the MP algorithm. For each iteration, the input signal which enters the iteration, chosen atom, and the difference between input signal and chosen atom are shown. At the bottom, the input signal reconstruction made of chosen atoms is shown.

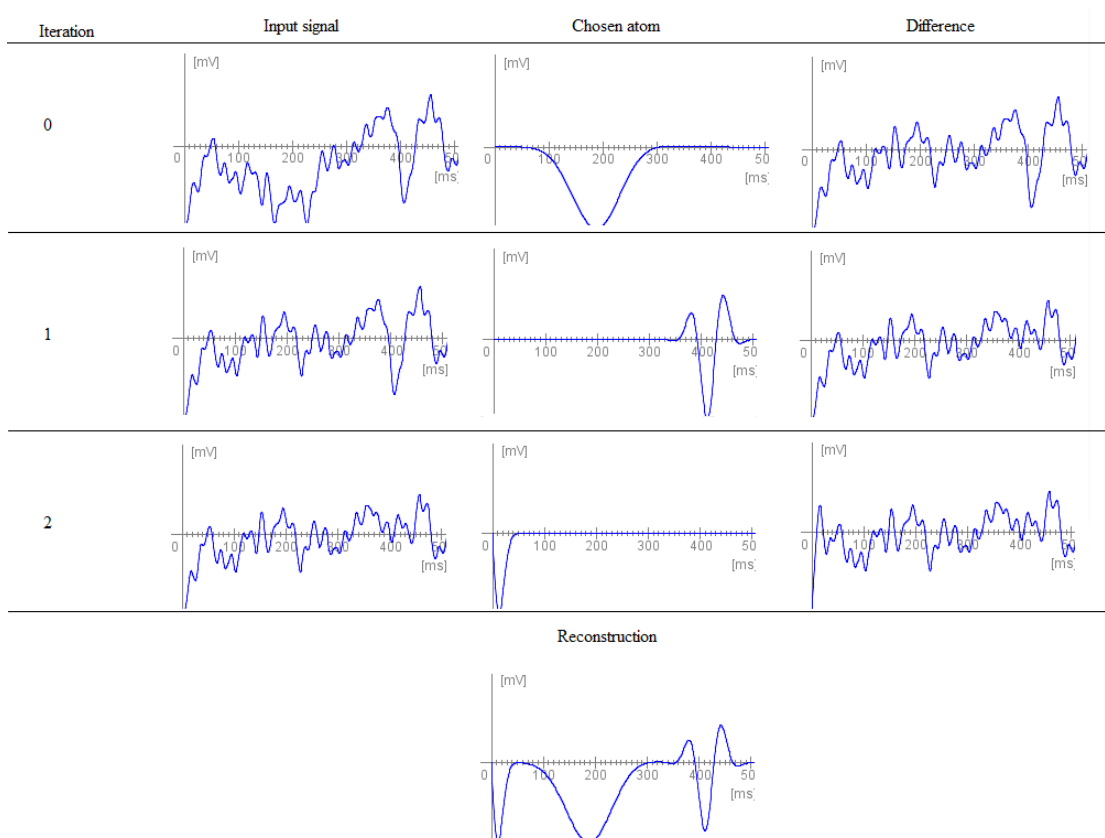


Figure 28: Example of three iterations of MP algorithm

#### 5.4.2 Output visualization

The result of MP algorithm is a two-dimensional matrix. Row dimension represents number of Gabor atoms. Column dimension is composed from all parameters of a Gabor atom: scale, shift, frequency and phase shift. This form of output is suitable for a next algorithmic processing because it contains all necessary information, but it is not suitable as a final output for scientists. For this purposes, the Wigner-Ville transform is usually used, which allows seeing important information of the MP algorithm output by a naked eye.

Wiener–Khinchin theorem says that the power spectral density of the signal  $x$  (which equals to Fourier transform of signal  $x$  squared) can be calculated as a Fourier transform of autocorrelation of the signal  $x$  [41]:

$$|\mathcal{F}(x)|^2 = \mathcal{F} \left( \int_{-\infty}^{+\infty} x \left( t + \frac{\tau}{2} \right) x \left( t - \frac{\tau}{2} \right) d\tau \right) \quad (25)$$

where the Fourier transform of the signal  $x$  is given by:

$$\mathcal{F}(x) = \int_{-\infty}^{+\infty} e^{-i\omega t} x(t) dt \quad (26)$$

Substituting equation (26) into equation (25) we get the following formula for the power spectral density of  $x$ :

$$|\mathcal{F}(x)|^2 = \int_{-\infty}^{+\infty} e^{-i\omega\tau} \left( \int_{-\infty}^{+\infty} x\left(t + \frac{\tau}{2}\right) x\left(t - \frac{\tau}{2}\right) dt \right) d\tau \quad (27)$$

If we remove the middle integral, corresponding to the integration over time, we get a time-dependent spectral density as a two-dimensional function:

$$W(x) = \int_{-\infty}^{+\infty} e^{-i\omega\tau} x\left(t + \frac{\tau}{2}\right) x\left(t - \frac{\tau}{2}\right) d\tau \quad (28)$$

which is the Wigner-Ville transform of  $x$ . This transform exhibits several elegant and desirable mathematical properties, hence it is sometimes considered a fundamental time-frequency representation. Unfortunately, it has also one major drawback which is the presence of cross-components in the time-frequency plane, as illustrated in Figure 29. [41]

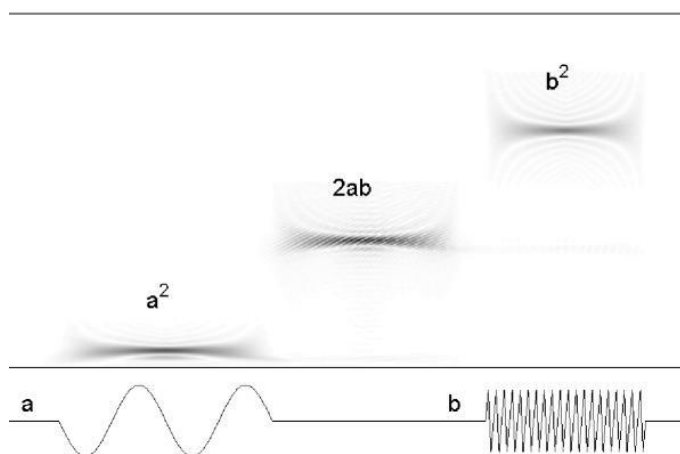


Figure 29: The cross-component is labeled by 2ab. [41]

Minimization of the presence of cross-terms in time-frequency estimates can be achieved by using such a kernel that the cross-terms won't appear.

Wigner-Ville distribution computed directly from equation (20) gives:

$$W(x) = W\left(\sum_{n=0}^{N-1} a_n g_n\right) = \sum_{n=0}^{N-1} a_n^2 W(g_n) + \sum_{n=0}^{N-1} \sum_{k=1; k \neq n}^{N-1} a_n a_k W(g_n, g_k) \quad (29)$$

where  $a_n = \langle R^n x, g_n \rangle$ . The double sum causes occurrence of the cross-components. If we omit this double sum, we get the Wigner-Ville transform without unwanted cross-components.

#### 5.4.3 ERPs detection with MP algorithm

The ERPs detection with MP algorithm is based on decomposition of the EEG/ERP signal into atoms. The main idea is that during this decomposition, according to

information about basic principle given above, the signal trend is approximated by a few first atoms whereas the signal details are approximated by the next atoms in later iterations. And it is just ERP waveform which should be a significant part of the signal trend.

According to equation (22) each atoms is defined by a ordered quadruple  $(s, u, v, w)$ . This quadruple enables to evaluate the Gabor atom at any time (for each value of parameter  $t$ ) and do its Wigner-Ville transform. When we are trying to detect an ERP waveform, we know its typical latency, so we will be looking for an atom which position corresponds with ERP's latency and which approximates well the trend of the signal – its power spectral density is significantly high.

Figures 30, 31, and 32 demonstrates described method for the P3 waveform detection. The input signal which contains the P3 waveform on position 300 ms, which can be easily seen, was approximated by twenty Gabor atoms.

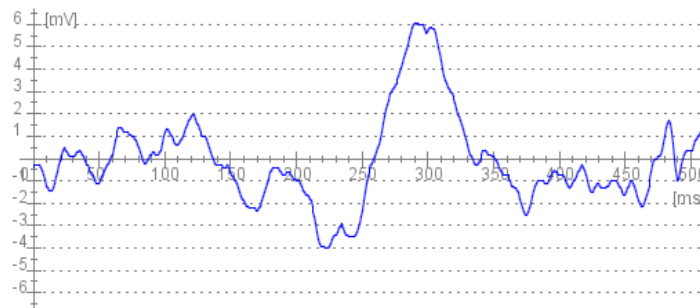


Figure 30: Input signal with P3 waveform. [20]

The atom which best approximates the P3 waveform is shown in Figure 31. It is also the atom selected in first iteration of the MP algorithm.

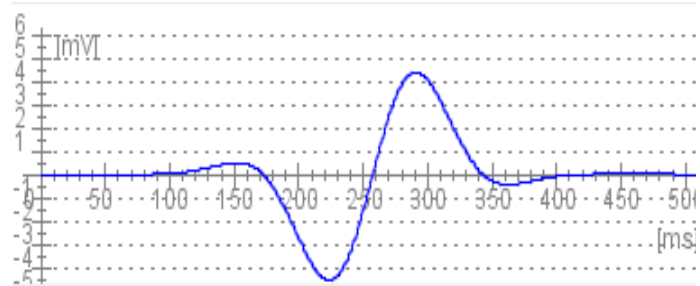


Figure 31: The atom which best approximates the P3 waveform. [20]

Now, it is time to use the Wigner-Ville transform which is able to show whether the power spectral density of this atom is high enough to prove the P3 waveform occurrence in the input signal.



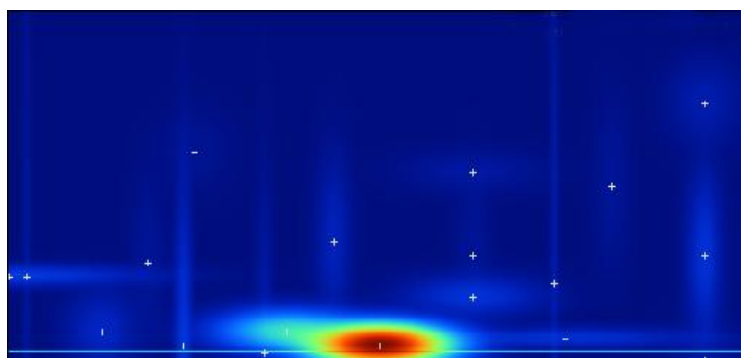


Figure 32: Wigner-Ville transform of 20 Gabor atoms which approximates the input signal. [20]

The Wigner-Ville transform shows all twenty Gabor atoms the input signal was decomposed by MP algorithm into. The significant area with high power spectrum density at the bottom on the middle in Figure 32 is the Wigner-Ville transform of the Gabor atom from Figure 31. Its area of occurrence corresponds with typical P3 waveform latency. This idea for ERP waveforms detection using the MP algorithm was published in [42].

#### 5.4.4 ERPs detection issues

However, the ERP waveform is not always approximated by one Gabor atom only. There are two different examples of the P3 component detection in Figure 33. On the left half of the figure, the favorable situation is shown - the P3 waveform is approximated by one Gabor atom only and the value of the correlation between this atom and the input signal is high enough to pass the threshold. On the right half of the figure, the unfavorable situation is shown. The P3 waveform is partially approximated by two Gabor atoms - the first one and the third one. The value of the correlation between the EEG/ERP signal and both the first and third Gabor atom is not high enough to pass the threshold. It leads to a false negative detection result.

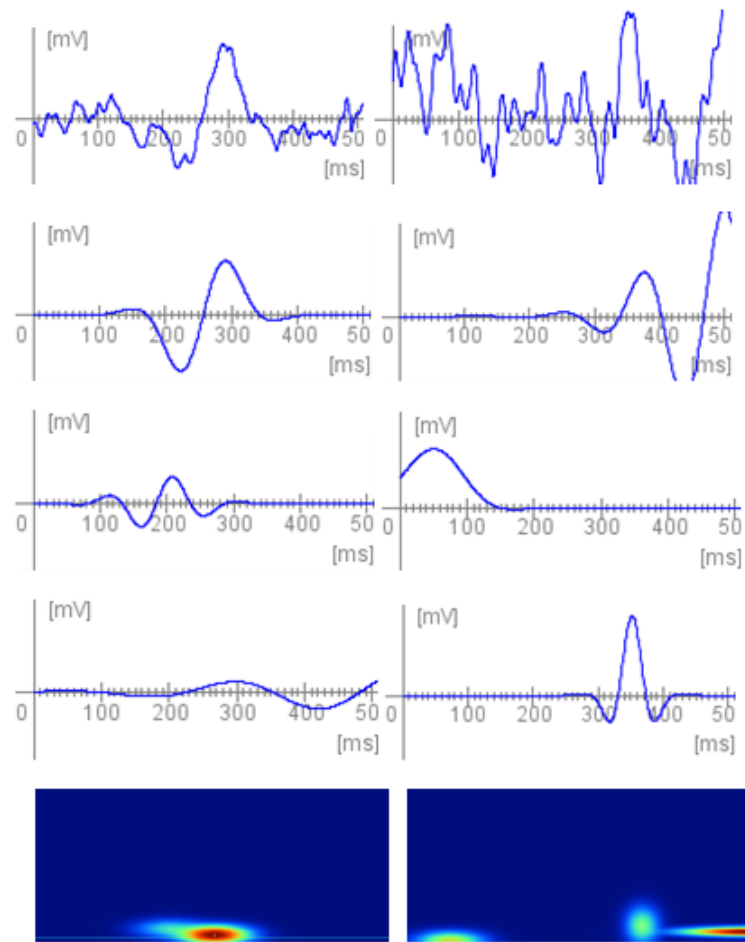


Figure 33: The favorable decomposition is shown on the left half and the unfavorable decomposition is shown on the right. In order from top to bottom: the input signal with the P3 waveform; the first, second, and third Gabor atom; visualization of Gabor atoms by the Wigner-Ville transform.

There are two ideas to solve this issue:

1. If we could select all atoms which partially approximate the ERP waveform, calculate the vector sum of these atoms and consider this vector sum as a new atom, we would be able to detect the ERP waveform successfully. The solution is to use an algorithm (e.g. self-organizing map) which categorizes atoms into groups in such a way that atoms in each group are similar to each other. Once we have these groups, we can manually mark the groups which contain atoms which can approximate (or partially approximate) ERP waveforms.
2. Do the input signal reconstruction from the Gabor atoms and then compute correlation between the reconstructed signal and a model of the ERP waveform [43]. This modification is described in the next chapter.

#### **5.4.4.1 Modification of ERP waveforms detection**

The basic idea is use the MP algorithm as the method of input signal filtering and then to compute correlation between filtered (reconstructed) signal and an ERP waveform model. [20]

First, we approximate the input signal (Figure 34) using several Gabor atoms and then we reconstruct the input signal from them. Loss of information caused by approximation is considered as filtering of the input signal (Figure 35). [36]

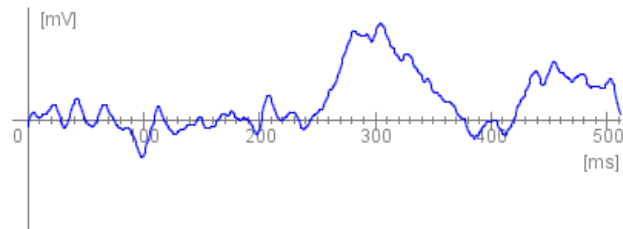


Figure 34: The input signal. [20]

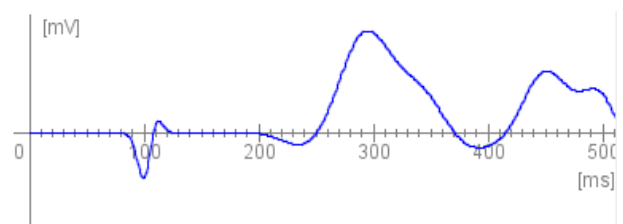


Figure 35: Reconstruction of the input signal from five Gabor atoms. [20]

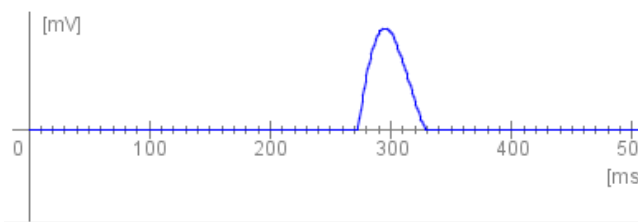


Figure 36: ERP waveform model in the corresponding location. [20]

The following phase includes the detection itself when an ERP waveform model (Figure 36) is used. This model is obtained e.g. by averaging a sufficient number of epochs containing raw ERP signal or by filtering of ERP waveform from one epoch. The ERP waveform model is shifted on the restored signal in the expected range of ERP waveform. Correlation between the ERP waveform model and the reconstructed signal is computed for each shift. A maximum value of the correlation and the attaching shift value are stored. After calculating all possible correlations the stored maximum value is compared to the threshold. If the maximum value is equal to or greater than the threshold, ERP waveform is detected in the corresponding location. [20]

## 5.5 Hilbert-Huang transform

The Hilbert-Huang transform (HHT) was designed to analyze nonlinear and non-stationary signal [22]. In fact, the HHT is composed of two independent transformations:

1. Empirical mode decomposition (EMD)
2. Hilbert spectral analysis (HSA)

In a nutshell, the EMD approximates the input signal by a sum of intrinsic mode functions (IMF). After the decomposition, each IMF is analyzed by HSA which determines frequency and amplitude at each functional value of the IMF function.

### 5.5.1 Intrinsic mode function

An intrinsic mode function (IMF) is a function which fulfills both following conditions:

1. In the whole data set, the number of extrema and the number of zero crossings must be either equal or differ by one at most [22].
2. The mean value of the envelope defined by the local maxima and the local minima is zero at any point [29; 30; 31].

An IMF represents simple oscillatory mode as counterpart to a simple harmonic function, but it is much more general by its definition. The conditions which IMF fulfills are necessary for defining instantaneous frequency [22].

### 5.5.2 Empirical mode decomposition

The goal of the empirical mode decomposition is to decompose the input signal to the IMFs and the residuum. The EMD is a data driven method and IMFs are derived directly from the signal itself [32]. Given an input signal  $x(t)$ , the effective algorithm of EMD – known as sifting - can be summarized as follows[29; 32; 33]:

1. Two smooth splines are constructed connecting all the maxima and minima of  $x(t)$  to get its upper envelope,  $Max(t)$  and its lower envelope,  $Min(t)$ ; The extrema can be simply found by determining the change of sign of the derivative of the signal. Once the extrema are identified, all the maxima are connected by a cubic spline line as the upper envelope. The procedure is repeated for the local minima to produce the lower envelope. All the data points should be covered by the upper and lower envelopes.
2. Compute the mean  $m(t) = \frac{Min(t) + Max(t)}{2}$
3. Extract the detail  $d(t) = x(t) - m(t)$
4. The process is repeated for  $d(t)$  until the resulting signal, the first IMF ( $imf_1(t)$ ), satisfies the criteria of an intrinsic mode function.

The residue  $r_1(t) = x(t) - imf_1(t)$  is then treated as new data subject to the sifting process as described above, yielding the second IMF from  $r_1(t)$ . The procedure continues until either the recovered IMF or the residual data are too small, in the sense that the integrals of their absolute values or the residual data have no turning points. Once all of the wavelike IMFs are subtracted from the data, the final residual component represents the overall trend of the data. [32]

By construction, the number of extrema is decreased when going from one residual to the next, and the whole decomposition is guaranteed to be completed with a finite number of modes. At the end of the described process, the input signal  $x(t)$  can be described by following expression:

$$x(t) = \sum_{j=1}^N imf_j(t) + r_N(t) \quad (30)$$

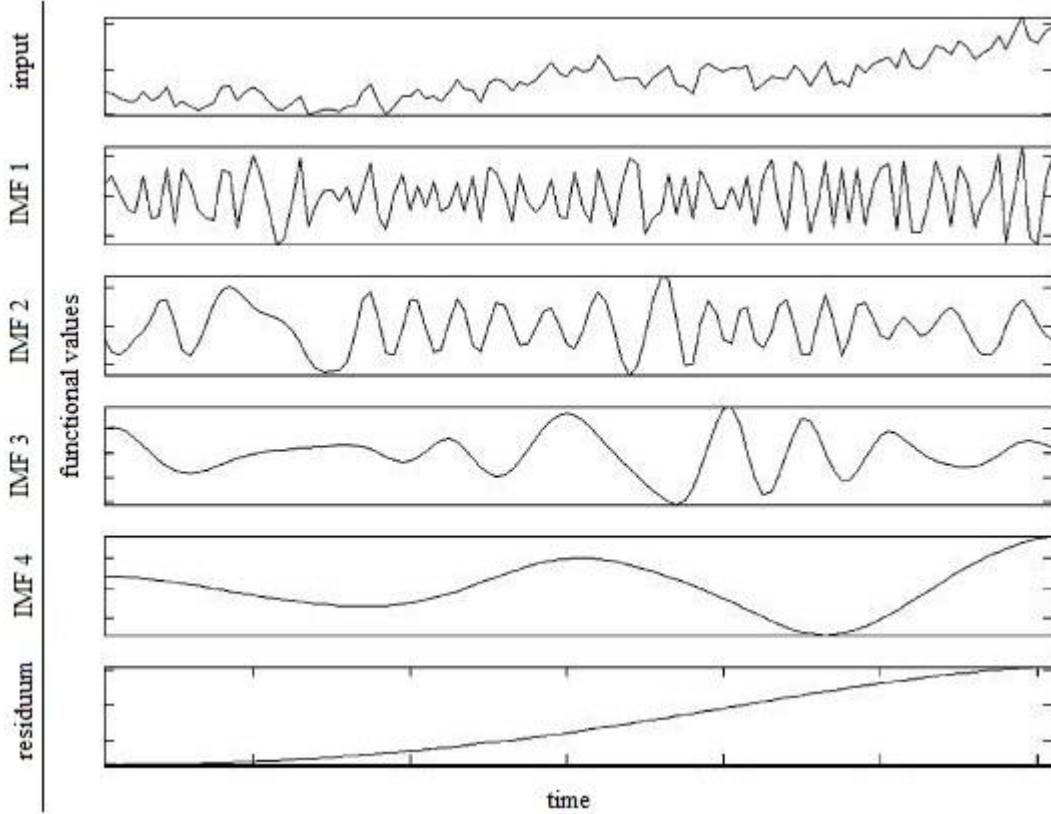


Figure 37: EMD example

### 5.5.3 Stopping criteria

The extraction of a mode is considered as satisfactory when the sifting process is terminated [32], it means that the two conditions mentioned in Chapter 5.5.1 must be fulfilled. The first one – that the number of extrema and the number of zero crossings must be either equal or differ by one at most – is relatively easy to fulfill. The second one – that mean of the envelopes is meant to be zero – is very difficult to fulfill [22]. Two stopping criteria were proposed. The first one is standard deviation (SD) [29; 34]:

$$SD = \sum_{t=0}^T \frac{|h_{k-1}(t) - h_k(t)|^2}{h^2_{k-1}(t)} \quad (31)$$

The SD can lead to unsatisfying result in case that for some value of  $t$  the value of the denominator is too close to zero. This situation could cause that the value of SD will practically be equal to infinity. Because of this reason, the usage of Cauchy convergence test (CCT) was demonstrated in [35] as an alternative to SD:

$$CCT = \frac{\sum_{t=0}^T |h_{k-1}(t) - h_k(t)|^2}{\sum_{t=0}^T h^2_{k-1}(t)} \quad (32)$$

#### 5.5.4 Hilbert transform

The Hilbert transform returns the analytic signal from real data sequence. The analytic signal:

$$x = x_r + i \cdot x_i \quad (33)$$

has its real part, which represents the original data, and its imaginary part, which contains the Hilbert transform. The imaginary part is a version of the original real sequence with a 90° phase shift. The Hilbert transformed series has the same amplitude and frequency content as the original real data and includes phase information that depends on the phase of the original data. The Hilbert transform is useful for calculating instantaneous attributes of time series, especially the amplitude and frequency. [36]

The Hilbert transform is useful for calculating instantaneous attributes of time series, especially the amplitude and frequency. The instantaneous amplitude is the amplitude of the complex Hilbert transform; the instantaneous frequency expresses the rate of change of the instantaneous phase angle. [22]

#### 5.5.5 Standard discrete-time analytic signal

The analytic signal for a sequence has a one-sided Fourier transform (with no negative frequencies). To approximate the analytic signal, the Hilbert method calculates a FFT of the input sequence, replaces those FFT coefficients corresponding to negative frequencies with zeros, and calculates an inverse FFT of the result [22]. Hilbert transform uses following algorithm [37, 22]:

1. FFT of the input sequence is calculated. The result is stored into a vector  $x$ .
2. A vector  $h$  with length  $n$  indexed by index  $i$  with following values is created:
  - 1 for  $i = 1, (n/2)+1$
  - 2 for  $i = 2, 3, \dots, (n/2)$
  - 0 for  $i = (n/2)+2, \dots, n$
3. An element wise product of  $x$  and  $h$  is stored in  $e$ .
4. The inverse FFT of the  $e$  is calculated and the first  $n$  items are returned as a result.

#### 5.5.6 Determining information about frequencies and amplitudes

When the EMD process is done, it is possible to calculate the analytic signal by using the algorithm described in Chapter 5.5.5. Calculation of the analytic signal  $Z(t)$  is defined as follows [28; 22; 31]:

$$Z(t) = X(t) + iY(t) = a(t)e^{i\theta(t)} \quad (34)$$

where  $X(t)$  is the original signal, and  $Y(t)$  is the Hilbert transform of  $X(t)$ . Then, the instantaneous attributes of  $Z(t)$  are defined:

$$a(t) = \sqrt{X(t)^2 + Y(t)^2} \quad (35)$$

$$\theta(t) = \arctan\left(\frac{Y(t)}{X(t)}\right) \quad (36)$$

$$\omega(t) = \frac{d\theta(t)}{dt} \quad (37)$$

where  $a(t)$  is the instantaneous amplitude,  $\theta(t)$  is the instantaneous phase and  $\omega(t)$  is the desired instantaneous frequency (see Figure 38 for an example of Hilbert spectrum visualization) [22].

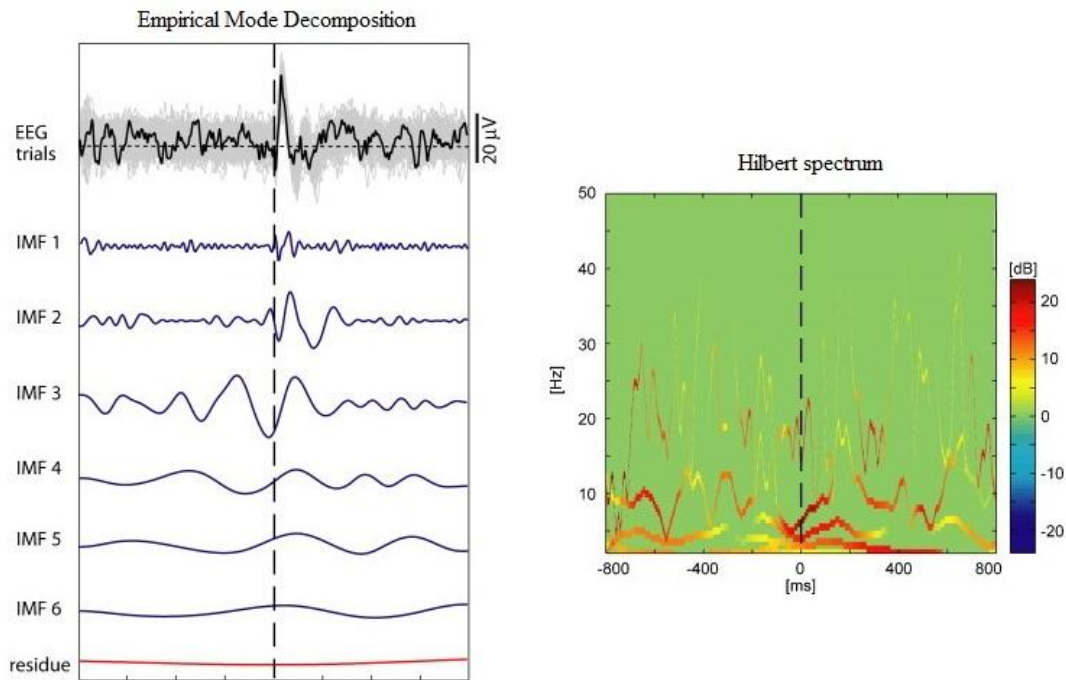


Figure 38: Hilbert spectrum visualization. [38]

### 5.5.7 ERPs detection with HHT

The basic idea of ERPs detection is based on knowledge of ERPs' typical frequencies and latencies. Both frequencies and latencies of waveforms of which the input EEG/ERP signal is composed from, does not disappear during the EMD process. They are only decomposed into IMFs – including frequencies and latencies ERPs are made of.

The first step is to decompose the EEG/ERP signal to IMFs using EMD. Then, the Hilbert transform is calculated for each IMF. Each IMF's Hilbert transform is searched for frequency and amplitude typical for the detecting ERP using method presented in Chapter 5.5.6 in region of estimated prevalence of the ERP.

## 5.6 Artificial neural networks

An artificial neural network (ANN) is an information-processing system that has certain performance characteristics in common with biological neural networks. ANNs have been developed as generalization of mathematical models of human cognition or neural biology, based on the assumption that: [44]

1. Information processing occurs at many simple elements called neurons.
2. Signals are passed between neurons over connection links.
3. Each connection link has an associated weight, which, in a typical neural net, multiplies the signal transmitted.
4. Each neuron applies an activation function (usually nonlinear) to its net input (sum of weighted input signals) to determine its output signal.

### 5.6.1 Mathematical model of artificial neural networks

The basic unit of any ANN is a formal neuron, which is a simplified mathematical description of a real neurophysiological neuron. Its structure is shown in Figure 39:

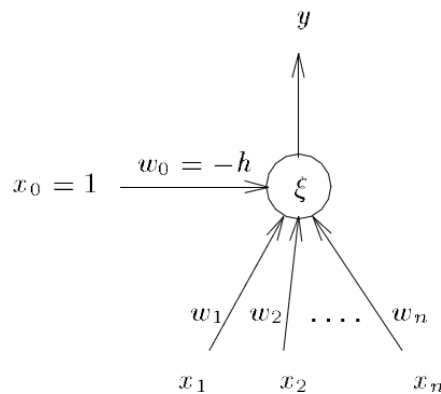


Figure 39: The mathematical model of a formal neuron. [45]

Generally speaking, the formal neuron is composed from:

- Vector  $[x_1, \dots, x_n]$  where  $x_i$  is a model of dendrite
- Vector  $[w_1, \dots, w_n]$  where  $w_i$  is a synaptic weight
- Inner potential  $\xi$  which is defined as follows:

$$\xi = \sum_{i=1}^n w_i x_i \quad (38)$$

- $h$  is a threshold value.
- $y$  is the output value

For purpose of generalization, the threshold value is represented by negative synaptic weight  $w_0$  which belongs to the constant input  $x_0$ . In this case, the sum in equation (38) starts with 0. The output  $y = \sigma(\xi)$  is activated when the inner potential reaches the threshold value:

$$y = \sigma(\xi) = \begin{cases} 1 & \text{for } \xi \geq 0 \\ 0 & \text{for } \xi < 0 \end{cases} ; \xi = \sum_{i=0}^n w_i x_i \quad (39)$$



## 5.6.2 Typical architectures

It is convenient to visualize neurons as arranged in layers (typically, the first layer is an input layer and the last layer is an output layer). Within each layer, neurons usually have the same activation function and the same pattern of connections to other neurons. To be more specific, in many neural networks, the neurons within a layer is connected to a neuron in another layer, then each hidden unit is connected to every output neuron. [44]

To arrangement of neurons into layers and the connection patterns within and between layers is called the net architecture [44]. The basic determination of ANN is to single-layer ANN and multilayer (note that the input layer is not counted as a layer).

### 5.6.2.1 *Single-layer*

A single-layer network has one layer of connection weights. Often, the units can be distinguished as input units, which receive signals from the outside world, and output units, from which the response of the net can be read. In the typical single-layer net, the input units are fully connected to output units, but are not connected to other input units, and the output units are not connected to other output units. [44]

### 5.6.2.2 *Multilayer*

A multilayer network is a net with one or more layers between the input layer and the output layer. Typically, there is a layer of weights between two adjacent layers. Multilayer nets can solve more complicated problems than single-layer nets can. [44]

## 5.6.3 Learning

Before we start using ANN for a classification, it is necessary to adapt it – found values of all weight vectors in such a way that the ANN is able to do a successful classification. Looking for a suitable weight vectors is called learning [46]. There are two ways to learn the ANN:

### 5.6.3.1 *Unsupervised learning*

Self-organizing neural networks group similar input vectors together without the use of training data to specify what a typical member of each group looks like or to which group each vector belongs. A sequence of input vectors is provided, but no target vectors are specified. The net modifies the weights so that the most similar input vectors are assigned to the same output unit (the term *cluster* is often used). [44]

### 5.6.3.2 *Supervised learning*

In perhaps the most typical neural network setting, training is accomplished by presenting a sequence of training vectors, or patterns, each with an associated target output vector [44]. A sequence of input vector is provided as well as information to which group each vector belongs to. During the learning process, the ANN does classification of each input vector separately. In case that the input vector is classified into the right class, the weights are not modified. In the other case, the weights are modified in such a way that the input vector will be classified into the right class next time.

#### 5.6.4 Artificial neural networks vs. ERP waveforms detection

ANN are very useful for solving tasks which are too complicated that it is not possible to describe all necessary relations by an exact mathematical model, or in case that this model is too complicated and its algorithmization is practically impossible [49]. The feature which allows using ANN in these cases is just the learning process:

*Learning is the significant feature of neural networks. This fact clearly expresses the main difference between currently used algorithms and problem-solving approaches based on neural networks. So far we have been putting all our effort into creating strict rules for transformation of input data into output data. Because of neural networks, we do not need to create the strict rules anymore. The “rules“ are created by neural networks during the learning process.* [48]

ANNs are suitable for ERP detection because of the nature of ERP waveforms. There is a theoretical description how the ERP waveforms should look like, how fast are they typical frequencies, how high are their typical amplitudes, and many milliseconds it typically takes when the ERP waveform appears after stimulus. But in praxis, every single ERP waveform obtained from every single measured subject at any time is a little bit different from each other previously obtained ERP waveform. Defining some exact rules for ERP waveform detection is a not trivial task and often must be done for each measured subject and for each scenario.

This is a good opportunity to use an ANN for ERP waveforms detection. But before we can start with learning, we have to do three steps:

1. Obtain training data set and split it to target / non-target epochs.
2. Choose a suitable feature vector.
3. Found right values for all parameters of ANN to classify the ERP waveforms well.

When the ANN is well learned, the ERP waveform detection process using ANN can start as shown in Figure 40.

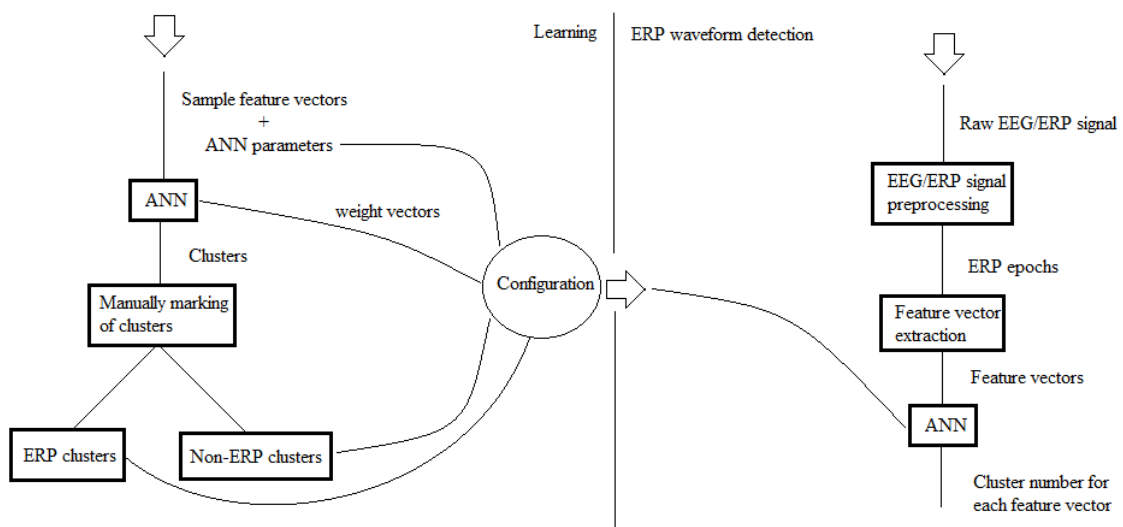


Figure 40: The unsupervised learning process and ERP waveforms classification process of a general ANN.

In case of supervised learning process, the **Manual marking of clusters** phase is substituted by a supervisor's control.

### 5.6.5 Feature vectors

The feature vector is defined as an output of an algorithm which extracts all significant information from signals which will be classified by an ANN. The only restriction of the feature vector is that number of its values has to be equal to number of neurons in the input layer of the ANN.

Choosing a suitable feature vector is a critical decision for further successful clustering. There are no exact or universal rules to identify optimal feature vector. In following list, there are some examples of feature vectors used in EEG domain:

- Subasi and Ercelebi used lifting-based discrete wavelet transform (LBDWT) coefficients of EEG signals as an input of classification system with two discrete outputs: epileptic seizure or non-epileptic seizure [52].
- Lotte et al. used a raw EEG signal, and amplitude values of smoothed EEG in [53]. Pradhan et al. used a raw EEG signal in [54], too.
- Gotman and Wang used average EEG amplitude, average EEG duration, coefficient of variation, dominant frequency, and average power spectrum in [55].

In experiments at our department, we tested following feature vectors:

- Result of DWT.
- Following feature vectors based on Gabor atoms:
  - All functional values of Gabor atom.
  - Functional values of Gabor atom subsampled to 32 samples.
  - Parameters of Gabor atom (scale, shift, frequency, and phase).
  - Feature vector which is composed of Gabor atom wave energy and wave local maximum/minimum. This feature extraction is shown in Figure 41. Note that wave energy is multiplied by  $-1$  in case that the wave is below the time axis.

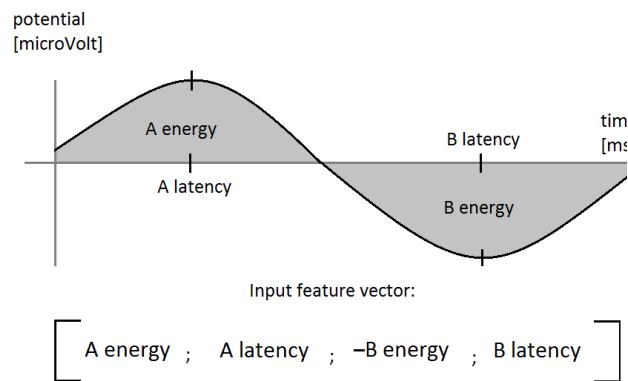


Figure 41: Transformation of Gabor atom into feature vector.

### 5.6.6 ART2 neural network

The ART (Adaptive Resonance Theory) network developed by Carpenter and Grossberg [12] is based on clustering. Its output is direct information about an output class. There are several ARTs (ART1, ART2, ARTMAP) differing by architecture and input feature vector type (binary or real valued) they are able to process [50]. For ERP waveforms detection, the ART 2 network, processing real-valued feature vector is suitable – it is a single-layered network with unsupervised learning.

The ART2 topology in Figure 42 contains three kinds of neurons (for detailed description of ART2 neural network architecture see [44]):

- Neurons of F1 layer which is called input layer. Each neuron represents one feature in feature vector.
- Neurons in F2 layer which is called output layer. Each neuron is related to one cluster.
- Neurons in R layer. This group of neurons creates so-called reset mechanism.

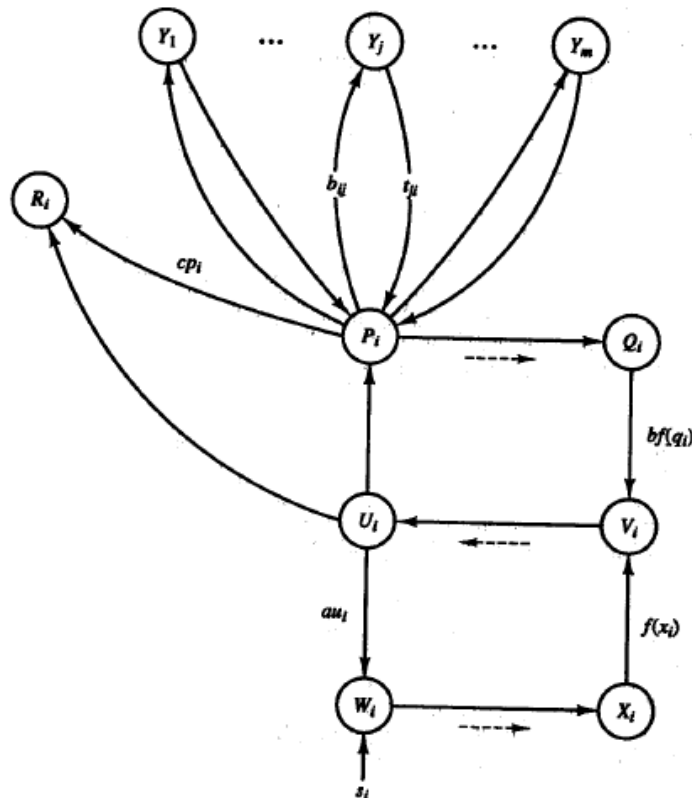


Figure 42: AT2 architecture [51]

The ART2 learning algorithm is divided to so-called epochs. In each epoch, all feature vectors are submitted one by one to input layer. Each feature vector comes from the F1 layer to the F2 layer. The F2 layer is competitive, i.e. all neurons in this layer compute response to the feature vector coming from the F1 layer. The neuron which response is the highest is marked as a winner. Now, it is the responsibility of the reset mechanism whether the winner's weight vector will be adapted on the feature vector or not. The

reset mechanism computes the similarity between the winner and the feature vector. If the similarity value is lower than a threshold value defined by user, then the competition starts again but without the winner. Otherwise, the winner is adapted on the feature vector (for more information about the adaptation process see [44]). [57]

It is important to note that the weight vectors values are sensitive to order of feature vectors during learning process [46]. It means that in case the feature vectors will be the same during the learning process, but will be submitted to the input layer in different order, the values of the weight vectors will be different.

#### 5.6.6.1.1 ERP waveforms detection with ART2

The ART2 is a suitable ANN for ERP waveforms detection according to schema in Figure 40 because:

- ART2 is a suitable ANN for data clustering [65]
- It has unsupervised learning algorithm. It means that it is not necessary to mark each feature vector from learning set as target/non-target. Instead of it, the whole clusters will be marked as target/non-target when the network will be well-learned. This corresponds with assumption, that the unsupervised learning algorithms are applied when the classification of a given set of sample patterns is unknown or not available [65].
- The unsupervised learning algorithm does not exhibit truly unsupervised learning capabilities in the sense that the number of classes in data must be specified in advance [65]. This is a big plus because in case waveforms in one cluster are not similar enough we can increase the number of clusters. On the other hand, in case those waveforms which should be in one cluster will be present in two or more clusters, we decrease the number of clusters [43].

Weight vector for each cluster is set when the ART2 is well-learned. To have an idea how typical waveforms of each cluster look like it is possible to show the learning feature vectors which belongs to each cluster. To demonstrate this option, we used the matching pursuit algorithm with Gabor atoms dictionary to preprocess an EEG/ERP signal. This signal contained the P3 waveforms. Gabor atoms subsampled to 32 samples were used as feature vectors. In Figure 43, the cluster which contains the waveforms which contains peaks corresponding to the P3 latency and amplitude is shown.

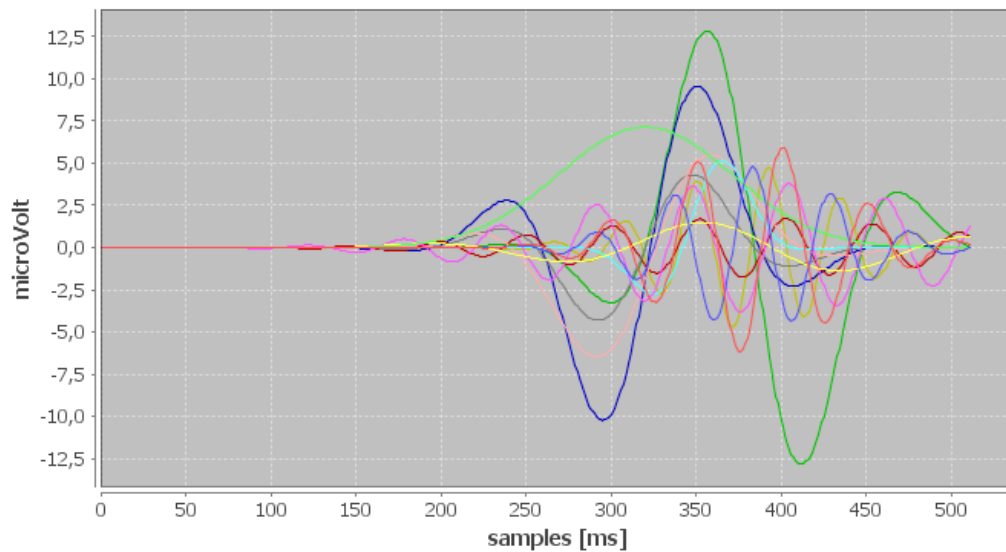


Figure 43: P3 waveform cluster

Now, it is possible to start with P3 waveforms detection. The Gabor atoms are submitted one-by-one to ART2's input layer. The P3 waveform is detected in case that a Gabor atoms is classified into the cluster which is shown in Figure 43.

#### 5.6.7 Self-organizing map (Kohonen map)

This chapter about self-organizing map (SOM) is taken from our paper [56]. There are  $N$  kinds of SOM topology where  $N$  is a dimension of a space where neurons are equidistantly placed.  $N$  is an integer value from the interval  $< 1; \infty$ ). For the ERP waveforms detection the most common solution is suitable: a two-dimensional organization of neurons. From now on, the SOM is considered as a two-dimensional map of neurons.

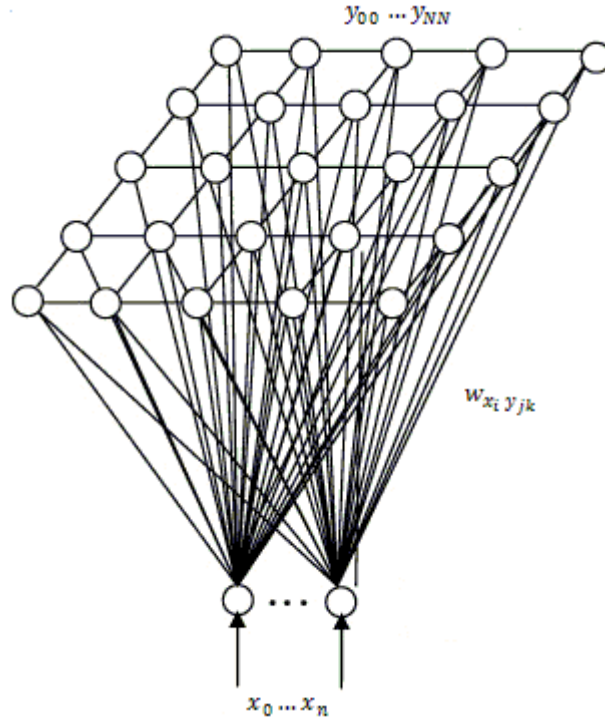


Figure 44: Schema of two-dimensional SOM architecture. [56]

The SOM is a one-layer network composed of neurons  $y_{jk}$ . Each neuron is connected with each item  $x_i$  in the input feature vector via a weighted connection  $w_{x_i y_{jk}}$ . Because the values from the input feature vector are directly forwarded to neurons SOM is a feed-forward neural network.

In the SOM, the winner weight vector and the weight vectors of all neurons in its neighborhood are modified during the learning process. It is necessary to choose a radius which defines size of the neighborhood. During experiments at our department, we used square neighborhood which radius was defined as follows:

$$radius = \left\lceil \frac{\alpha}{b^p} \cdot \frac{\sqrt{N^2}}{2} \right\rceil \quad (40)$$

where  $\alpha$  is the learning rate parameter,  $b$  is the base of exponential lost (the radius decreases with each next training pattern exponentially),  $p$  is the current learning progress

$$p = done/all \quad (41)$$

where  $done$  is the number of training patterns which were already used and  $all$  is the number of all training patterns, and  $N^2$  is total number of neurons.

As a result of the learning algorithm we get a specific weight vector for each neuron. It would be worth to have the set of weight vectors such that only one neuron would be marked as a winner for all similar atoms. Unfortunately, it does not work in the case of

the SOM because the selected atom weight vector and also weight vectors of all atoms in its neighborhood are updated during the learning process.

If we want to have all neurons with the similar weight vector in one cluster, we need a method to recognize these neurons and consider them as one cluster. At first, it is necessary to choose a metric for weight similarity. We decided to use a well-known method for measuring signal similarity – correlation. Equation (42) shows computing of correlation between two signals  $x$  and  $y$ :

$$crrltn(x, y) = \frac{\sum_{i=0}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=0}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=0}^n (y_i - \bar{y})^2}} \quad (42)$$

where:

$$\bar{x} = \frac{1}{n} \sum_{i=0}^n x_i \quad (43)$$

$$\bar{y} = \frac{1}{n} \sum_{i=0}^n y_i \quad (44)$$

### 5.6.7.1 Weight vectors similarity visualization

For better understanding, look at visualization of similarity between neuron weights where each neuron is shown as a 3x3 matrix. On the index  $[i-1, j-1]$  is the value of correlation between the neuron on the index  $[i, j]$  and the index  $[i-1, j-1]$ , etc. Note that there is always zero on the index  $[i, j]$ . For visualization, the values of the correlation result are recalculated from the interval of real values  $<-1, 1>$  to the interval of integer values  $<0, 255>$  (the gray scale).

$i-1, j-1$	$i-1, j$	$i-1, j+1$
$i, j-1$	$i, j$	$i, j+1$
$i+1, j-1$	$i+1, j$	$i+1, j+1$

Figure 45: Mask for visualization of weights similarity between neurons.

According to the description given above, visualization of weights similarity of neurons looks as shown in Figure 46.



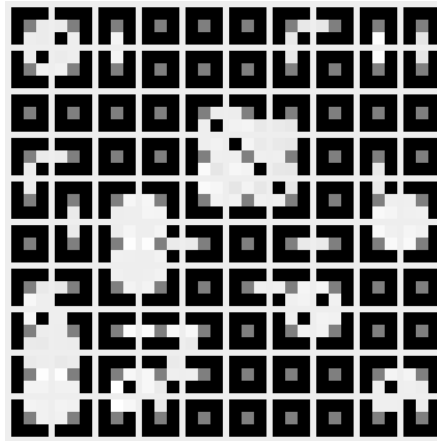


Figure 46: Neuron weights similarity in a two-dimensional map with 100 neurons.

It is easy to see clusters in Figure 46, but we need to implement an algorithm which is able to find these clusters as well. As a solution, we used an algorithm which is well-known in computer vision – connected-component labeling.

### 5.6.7.2 *Connected-component labeling*

Connected-component labeling (CCL) is a two-pass algorithm. It uses a map of neurons as an input. In the first pass, CCL iterates through each neuron by row. The neighboring neurons are given by a mask:

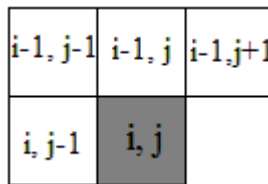


Figure 47: Mask which defines neighboring neurons during the first pass of the CCL algorithm.

According to correlation between the weight vector of the neuron  $[i, j]$  and weight vectors of neighboring neurons three situations can occur:

1. If correlation with all neighboring neurons is too low to be in the same cluster with neuron on the index  $[i, j]$  then a new cluster number is set to neuron on the index  $[i, j]$ .
2. If correlation of just one of neighboring neurons is high enough to be in the same cluster as neuron on the index  $[i, j]$  then the neuron on position  $[i, j]$  is put to the same cluster.
3. If correlation of more than one of neighboring neurons is high enough to be in the same cluster as neuron on the index  $[i, j]$  one of them is randomly selected and the neuron on position  $[i, j]$  is put to the same cluster. If neighboring neurons with high enough correlation value belong to different clusters, save that these clusters are equivalent to a special data structure.

In the second pass, CCL iterates through each neuron by row and gets rid of equivalent cluster numbers for one cluster using the special data structure from the first pass. After

the second pass, each cluster is signed with just one cluster number (even if the cluster consists of one neuron only).

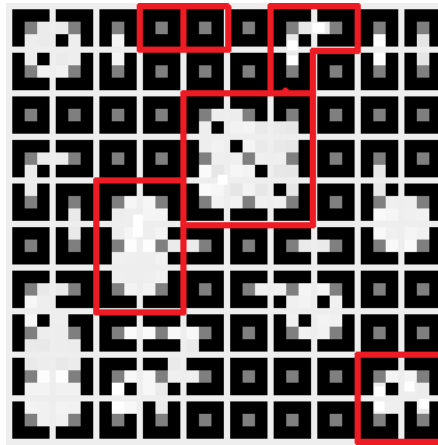


Figure 48: Neuron weights similarity in a two-dimensional map with 100 neurons with manually highlighted clusters which are related to Gabor atoms which approximate ERP P3 waveform.

### ***5.6.7.3 ERP waveforms detection with SOM***

The idea of ERP waveforms detection with SOM is quite simple. When the SOM is well-learned and clusters are identified by the CCL algorithm, it is possible to start with detection of an ERP waveform. According to SOM principle, for each feature vector a cluster is identified. If the cluster is a cluster which approximates the ERP waveform then the ERP waveform is detected. Otherwise the ERP waveform is not detected.

## 6 Scope of the Ph.D. Thesis

In this thesis, the overview of methods for ERP waveforms detection is given. Of course, it is practically not possible to describe all existing methods for ERP waveforms processing (e.g. following methods are suitable as well: principal component analysis (PCA), independent component analysis (ICA), support vector machines (SVM), etc.). This thesis describes at least one method from all mainstream approaches for ERP waveforms detection:

- methods based on statistical approach
- methods in time-frequency domain
- methods based on approximation/decomposition of the EEG/ERP signal
- methods based on artificial neural networks

### 6.1 Disadvantages of introduced methods

All proposed methods except of artificial neural networks have a common disadvantage: if they should work on any EEG/ERP signal obtained from any measured subject, then their classification rules must be general (intervals of frequencies, latencies, amplitudes, etc. expected from detecting ERP waveforms must be wide enough to cover all values which can appear). This approach leads to false positive detections (because the intervals are too wide). On the other hand, strict enabled values of ERP waveforms attributes lead to good detection result for one measured subject, but they lead to false negative detections. The strict enabled values are unusable for other all other subjects.

According to previous paragraph, it could look like the best solution for ERP waveforms detection is based on artificial neural networks. But ANNs have disadvantages depending on learning process:

- ANN with supervised learning: It is well-known that for successful learning of an ANN it is necessary to use a relatively large set of learning data (what is *large* depends on concrete domain). Obtaining a large set of ERP data from different subjects in a time-consuming task. But no extra skills are required and the experiment scenario for data acquisition is usually much less complicated than a scenario typically used in BCI systems. The problem with marking of learning data. This process takes very long time and only an expert in EEG/ERP domain (usually a neurologist) is able to mark the data right and divide them to appropriate classes.
- ANN with unsupervised learning: Advantage of the unsupervised learning is that we do not have to have an EEG/ERP expert for marking learning data. The ANN does clusters on its own – and there the disadvantage is hidden. We do not have a direct control over which learning feature vector will be assigned to which cluster. We can define number of neurons in output layer, we can define some ANN parameters (e.g. parameter of reset mechanism in ART2), but the direct control is still missing. As a result of the learning process, we get clusters which are suboptimal, but we are usually not able to fix it.

Looking at disadvantages mentioned above, we were looking for an ERP waveforms detection method which is suitable for use in BCI systems. One option seems to be the use of adaptive filters.

## 6.2 Adaptive filters

This adaptive filters characteristic is taken from [59]: An *adaptive filter* is defined as a self-designing system that relies for its operation on a recursive algorithm, which makes it possible for the filter to perform satisfactorily in an environment where knowledge of the relevant statistics is not available. Adaptive filters are classified into two main groups: linear, and non linear. *Linear adaptive filters* compute an estimate of a desired response by using a linear combination of the available set of observables applied to the input of the filter. Otherwise, the adaptive filter is said to be nonlinear. Adaptive filters may also be classified into:

- *Supervised adaptive filters*, which require the availability of a training sequence that provides different realizations of a desired response for a specified input signal vector. The desired response is compared against the actual response of the filter due to the input signal vector, and the resulting error signal is used to adjust the free parameters of the filter. The process of parameter adjustments is continued in a step-by-step fashion until a steady-state condition is established.
- *Unsupervised adaptive filters*, which performs adjustments of its free parameters without the need for a desired response. For the filter to perform its function, its design includes a set of rules that enable it to compute an input-output mapping with specific desirable properties. In the signal-processing literature, unsupervised adaptive filtering is often referred to as blind deconvolution or blind adaptation.

Adaptive filters find applications in highly diverse fields: channel equalization, system identification, predictive deconvolution, spectral analysis, signal detection, noise cancellation, and beamforming.

### 6.2.1 Adaptive filters and BCI – idea

The idea of adaptive filters use in BCI system is following: at first, the adaptation on preprocessed EEG signal (common activity) without any ERP will be done. The target of this adaptation is to learn the adaptive filter to suppress the EEG signal – it means, that on the output of the adaptive filter, there will be (in optimal case) constant zero function. Then, when an ERP waveform appears along with EEG activity on the input of the adaptive filter, there a typical response for the ERP waveform appear on output of the adaptive filter. At this time, it is not possible to say how this response will look like. But we expect that this response will be unique for each ERP waveform and it will be detectable by a suitable method. In the ideal case, the Ph.D. thesis will provide a suitable method for ERP waveforms detection which can be used in BCI systems, e.g. it responds in real time.

## 7 Conclusion

This thesis brought a brief introduction to BCI systems based on ERPs with emphasis on suitable methods for detection of ERP waveforms. At least one method from all mainstream approaches for ERP waveforms detection was described. Nowadays, the BCI systems are not widely used, but for a few people who need them, they work well. But there are still great opportunities for innovation:

- The time which is necessary for enter a one character with BCI spelling system based on the P3 waveform is quite long.
- There are approximately 3 % of people who are not able to use ERP based BCI systems at all.
- It is not comfortable to use a BCI based on SSVEP for a long time.
- The learning process of methods which need them (typically ANNs) takes a long time. An expert (usually a neurologist) is necessary to mark the training data set when the algorithm needs a supervised learning procedure.
- Computational complexity of some algorithms is at the edge or behind the edge for real-time signal processing (BCI desing has to obey, e.g. put as long time delays as needed between entering of two characters).

We will focus on the two last points mentioned above. We assume that solution could be use of adaptive filters. By definition, adaptation is much simpler process than learning and does not require supervision. To the computational complexity - we know from discrete signal processing that use of a filter is not a time-consuming task. Last, the kernel of an adaptive filter can be based on many different approaches/algorithms.

### 7.1 Aims of the Ph.D. thesis

- Identify or design EEG/ERP signal preprocessing methods suitable for adaptive filters.
- Identify or design adaptive filter(s) applicable in ERP waveforms detection process in BCI systems.
- Implement the selected adaptive filter(s) in a suitable programming language.
- Validate the proposed solution on a simple BCI based on ERPs or SSVEP.

## References

- [1] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurthsheller, and T. M. Vaughan. Brain-computer interfaces for communication and control, *Clinical Neurophysiology*, Volume 113, Issue 6, June 2002, pp. 767-791.
- [2] G. Schalk, P. Brunner, L.A. Gerhardt, H. Bischof, and J.R. Wolpaw. Brain-computer interfaces (BCIs): Detection instead of classification, *Journal of Neuroscience Methods*, Volume 167, Issue 1, 15 January 2008, pp. 51-62.
- [3] S. P. Levine, J. E. Huggins, S. L. BeMent, R. K. Kushwaha, L. A. Schuh, M. M. Rohde, E. A. Passaro, D. A. Ross, K. V. Elisevich, and B. J. Smith, A direct brain interface based on event-related potentials, *IEEE Transactions on Rehabilitation Engineering*, Volume 8, Issue 2, June 2000, pp. 180-185.
- [4] R. Sitaram, H. Zhang, C. Guan, M. Thulasidas, Y. Hoshi, A. Ishikawa, K. Shimizu, and N. Birbaumer. Temporal classification of multichannel near-infrared spectroscopy signals of motor imagery for developing a brain-computer interface, *NeuroImage*, Volume 34, Issue 4, 15 February 2007, pp. 1416-1427.
- [5] [BCI5] N. Weiskopf, R. Veit, M. Erb, K. Mathiak, W. Grodd, R. Goebel, and N. Birbaumer. 2003. Physiological self-regulation of regional brain activity using real-time functional magnetic resonance imaging (fMRI): methodology and exemplary data. *NeuroImage*, Volume 19, Issue 3, July 2003, pp. 577-586.
- [6] N. Weiskopf, K. Mathiak, S. W. Bock, F. Scharnowski, R. Veit, W. Grodd, R. Goebel, and N. Birbaumer., Principles of a brain-computer interface (BCI) based on real-time functional magnetic resonance imaging (fMRI). *IEEE Transactions on Biomedical Engineering*, Volume 51, Issue 6, June 2004, pp. 966-970.
- [7] N. Weiskopf, F. Scharnowski, R. Veit, R. Goebel, N. Birbaumer, and K. Mathiak, Self-regulation of local brain activity using realtime functional magnetic resonance imaging (fMRI). *Journal of Physiology (Paris)*, Volume 98, Issues 4-6, pp. 357-373.
- [8] J. R. Wolpaw, N. Birbaumer, W. J. Heetderks, D. J. McFarland, P. H. Peckham, G. Schalk, E. Donchin, L. A. Quatrano, C. J. Robinson, and T. M. Vaughan. Brain-computer interface technology: a review of the first international meeting, *IEEE Transactions on Rehabilitation Engineering*, Volume 8, Issue 2, 2000, pp. 164-173.
- [9] S. Sanei and J. A. Chambers. EEG Signal Processing, Chippenham (Wiltshire): Antony Rowe Ltd., 2007. ISBN 0470025816
- [10] N. K. Squires, K. C. Squires, and S. A. Hillyard. Two varieties of long-latency positive waves evoked by unpredictable auditory stimuli in man., *Electroencephalography and Clinical Neurophysiology*, Volume 38, Issue 4, April 1975, pp. 387-401.
- [11] D. Valbuena, I. Sugiarto, and A. Gräser. Spelling with the brain-computer interface and the integrated SSVEP stimulator, *Proceedings of the 4<sup>th</sup> International Brain-Computer Interface Workshop and Training Course*, 2008, pp. 291-296.
- [12] F. Vialatte, M. Maurice, J. Dauwels, and A. Cichocki. Steady-state visually evoked potentials: focus on essential paradigms and future perspectives, *Progress in Neurobiology*, Volume 90, Issue 4, April 2010, pp. 418-438.
- [13] 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*, Volume 2, Issue 1, October 2005, pp. 123-130.

- [14] C. Gugera, S. Dabana, E. Sellers, C. Holzner, G. Krausza, R. Carabalonac, F. Gramaticac, and G. Edlingera. How many people are able to control a P300-based brain-computer interface (BCI)?, *Neuroscience Letters*, Volume 462, Issue 1, 18 September 2009, pp. 94-98.
- [15] S. J. Luck. *An introduction to the Event-Related Technique*, The MIT Press, London, 2005. ISBN 0-262-12277-4
- [16] J. N. Knight. *Signal Fraction Analysis and Artifact Removal in EEG*. Ph.D. Thesis, Colorado State University, Fort Collins, USA, 2003.
- [17] Available on <http://www.ccs.neu.edu/groups/honors-program/freshsem/19951996/cloder/neuron.html> [cited 2010].
- [18] T. Hinterberger, R. Veit, B. Wilhelm, N. Weiskopf, J. J. Vatine, and N. Birbaumer. Neuronal mechanisms underlying control of a brain-computer interface, *The European Journal of Neuroscience*, Volume 21, Issue 11, June 2005, pp. 3169-3181.
- [19] A. Kubler, B. Kotchoubey, J. Kaiser, J. R. Wolpaw, and N. Birbaumer. Brain-computer communication: unlocking the locked in, *Psychological bulletin*, Volume 127, Issue 3, May 2001, pp. 358-375.
- [20] T. Rondik. *Methods of ERP Signal Processing*, Diploma Thesis. University of West Bohemia, Pilsen, Czech Republic, 2010.
- [21] T. C. Handy. *Event-Related Potentials: A Methods Handbook*. Cambridge, The MIT Press, 2004. ISBN 0-262-08333-7
- [22] J. Ciniburk. *Hilbert-Huang transform for ERP detection*, Ph.D. Thesis, University of West Bohemia, Pilsen, Czech Republic, 2011.
- [23] J.J. Vidal. Real-time detection of brain events in EEG. *Proceedings of the IEEE*, Volume 65, Issue 5, May 1977, pp. 633 - 641.
- [24] E. Jarosova and J. Reif. ANOVA (its robustness and power). *Proceedings of the 10<sup>th</sup> winter school JCMT*, 1998, pp. 89 - 102. ISBN 80-7015-624-4
- [25] W. Drongelen. *Signal Processing for Neuroscientists*. Academic Press, 2007. ISBN 0-12-370867-2
- [26] NI LabVIEW 8.6 manual, chapter STFT Spectrograms VI. Available on [http://zone.ni.com/reference/en-XX/help/371361E-01/lvanls/stft\\_spectrogram\\_core/](http://zone.ni.com/reference/en-XX/help/371361E-01/lvanls/stft_spectrogram_core/) [cited August 22, 2012].
- [27] S. W. SMITH. *The Scientist and Engineer's Guide to Digital Signal Processing*, San Diego (California): California Technical Publishing, 1997. ISBN 0-9660176-3-3
- [28] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N. C. Yen, C. C. Tung, and H. H. Liu. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis, *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, March 1998.
- [29] N. E. Huang and et al. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis, *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 1998.
- [30] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N. C. Yen, C. C. Tung, and H. H. Liu. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis, *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, March 1998.
- [31] R. Liu. Empirical mode decomposition: A useful technique for neuroscience?, *Proceedings of the Royal Society of London*, January 2002.

- [32] H Liang, S Bresser, and R Desimone. Empirical mode decomposition: a method for analyzing neural data. *Neurocomputing*, Volume 65-66, 2005, pp. 801-807.
- [33] G. Rilling, P. Flandrin, and P. Goncalves. On empirical mode decomposition and its algorithms, *Proceedings of the 6th IEEE/EURASIP Workshop on Nonlinear Signal and Image Processing*, Grado, Italy, 2003, pages 8-11, 2003.
- [34] F. Cong, T. Sipola, X. Xu, T. Huttunen-Scott, H. Lyytinen, and T. Ristaniemi. Concatenated trial based hilbert-huang transformation on event-related potentials, *The 2010 International Joint Conference on Neural Networks*, 2010, pp. 1-5.
- [35] N. E. Huang and N. O. Attoh-Okine. *The Hilbert-Huang Transform in Engineering*, CRC Press, 2005. ISBN: 978-0849334221
- [36] T. Rondik, J. Ciniburk. Comparison of various approaches for P3 component detection using basic methods for signal processing, *2011 4<sup>th</sup> International Conference on Biomedical Engineering and Informatics*, New York: IEEE, 2010, pp. 698-702. ISBN: 978-1-4244-9352-4
- [37] S. L. Marple. Computing the discrete-time 'analytic' signal via FFT, Conference Record of the Thirty-First Asilomar Conference on Signals, Systems & Computers, Volume 2, 1997, pp. 1322-1325.
- [38] A. Pigorini, A. G. Casali, S. Casarotto, F. Ferrarelli, G. Baselli, M. Mariotti, M. Massimini, M. Rosanova. Time-frequency spectral analysis of TMS-evoked EEG oscillations by means of Hilbert-Huang transform, *Journal of Neuroscience Methods*, Volume 198, Issue 2, 15 June 2011, pp. 236-245.
- [39] Balakrishnama, S., Ganapathiraju, A. Linear Discriminant Analysis – a brief tutorial, Technical report, Institute for Signal and Information Processing, Department of Electrical and Computer Engineering, Mississippi State University.
- [40] S. E. Ferrando, L. A. Kolasa, and N. Kovacevic. Algorithm 820: A Flexible Implementation of Matching Pursuit for Gabor Functions on the Interval. *ACM Transactions on Mathematical Software*, September 2002, Volume 28, Issue 3, pp. 337 – 353.
- [41] P. J. Durka. Matching pursuit, Available on [http://www.scholarpedia.org/article/Matching\\_pursuit](http://www.scholarpedia.org/article/Matching_pursuit) [cited April 22, 2010].
- [42] J. Svoboda, P. Mautner, and R. Moucek, Detection of ERP using Matching Pursuit Algorithm. Xth International conference on cognitive neuroscience, Bodrum: Lundbeck Ilac Ticaret Limited Sirketi, 2008, pp. 226.
- [43] T. Rondik, P. Mautner. Using ART2 for Clustering of Gabor Atoms Describing ERP P3 Waveforms, Not published yet.
- [44] L. Fausett. *Fundamentals of Neural Networks: Architectures, Algorithms and Applications*, Prentice Hall, 1993. ISBN 978-0133341867
- [45] J. Sima, R. Neruda. *Theoretical issues of neural networks*, Matfyzpress, Prague, 1996. ISBN 80-85863-18-9
- [46] K. S. Yuan. *Digital neural networks*. Prentice-Hall, New Jersey, 1993. ISBN 978-0136123262
- [47] E. Fiesler, R. Beale. *Handbook of neural computation (Computational Intelligence Library)*. IOP Publishing Ltd and Oxford University Press, 1997. ISBN 978-0750303125
- [48] I. Vondrak. *Artificial Intelligence and Neural Networks*. VSB-TU Ostrava, 2001, ISBN 80-7078-259-5
- [49] L. Vareka, L. Neural networks in ERP signal processing, Diploma Thesis, University of West Bohemia, Pilsen, 2010.



- [50] L. Vareka, P. Mautner. The Event-Related Potential Data Processing Using ART 2 Network, Not published yet.
- [51] G. A. Carpenter, S. Grossberg, ART2: self-organization of Stable Category Recognition Codes for Analog Input Patterns, *Applied Optics*, Volume 26, Issue 23, December 1987, pp. 4919-4930.
- [52] A. Subasi and E. Ercelebi, Classification of EEG signals using neural network and logistic regression. *Computer Methods and Programs in Biomedicine*, Volume 78, 2005, pp. 87–99.
- [53] F. Lotte, M. Congedo, A. Lecuyer, F. Lamarche, and B. Arnaldi, A Review of Classification Algorithms for EEG-based Brain-Computer Interfaces, *Journal of Neural Engineering*, Volume 4, Issue 2, January 2007.
- [54] N. Pradhan, P. K. Sadasivan, and G. R. Arunodaya, Detection of seizure activity in EEG by an artificial neural network: A preliminary study, *Computers and Biomedical Research, an international journal*, Volume 29, Issue 4, August 1996, pp. 303–313.
- [55] J. Gotman and L. Wang, State-dependent spike detection: Concepts and preliminary results, *Electroencephalography and Clinical Neurophysiology*, Volume 79, Issue 1, 1991, pp. 11–19.
- [56] T. Rondik, P. Mautner. Clustering of Gabor Atoms Describing Event-Related Potentials: Solution for ERP Detection Algorithm Based on Matching Pursuit when ERP Waveform is Approximated by Two or More Gabor Atoms, Not published yet.
- [57] M. Valenta. Czech document processing using neural networks, Diploma Thesis, University of West Bohemia, Pilsen, Czech Republic, 2011.
- [58] B. Widrow and R. Winter. Neural Nets for Adaptive Filtering and Adaptive Pattern Recognition, *Computer*, Volume 21, Issue 3, March 1988, pp. 25-39.
- [59] S. O. Hyakin. Adaptive Filter Theory (4<sup>th</sup> Edition), Prentice Hall, New Jersey, 2001. ISBN 978-0130901262
- [60] P. Soukal. Methods for automatic detection of ERP components, Diploma Thesis, University of West Bohemia, Pilsen, Czech Republic, 2010.
- [61] C. A. Valens. A Really Friendly Guide to Wavelets. Available on <http://www.robots.ox.ac.uk/~parg/mlrg/papers/arfgtw.pdf> [cited August 27, 2012].
- [62] A. Prochazka, E. Hostalkova. Biomedical signals and image processing using wavelet transform, *Automation: professional journal for automation, measurement and computer science engineering*, Volume 50, Issue 6, June 2007, pp. 397 – 401.
- [63] P. S. Addison. The Illustrated Wavelet Transform Handbook, IOP Publishing, Bristol, 2001. ISBN 0 7503 0692 0
- [64] S. Mallat. A wavelet tour of signal processing (Second edition). Academic Press, San Diego, 1999. ISBN 978-0-12-466606-1
- [65] Chin-Der Wann and S. C. A. Thomopoulos, A Comparative Study of Self-organizing Clustering Algorithms Dignet and ART2, *Neural Networks*, Volume 10, Issue 4, June 1997, pp 737–753.
- [66] J. C. Lee and D. S. Tan. Using a low-cost electroencephalograph for task classification in HCI research, *Proceedings of the 19th annual ACM symposium on User interface software and technology*, ACM New York, NY, USA, 2006, pp. 81-90.
- [67] A. Schlog; M. Slater; and G. Pfurtscheller. Presence Research and EEG, *5th International Workshop on Presence*, 2002.
- [68] Smith, R. C.: Electroencephalograph based Brain Computer Interfaces. Ph.D. Thesis, University College Dublin, Ireland, 2004.

## List of figures

Figure 1:	Schema of a nerve cell. ....	5
Figure 2:	Major brain waves. ....	7
Figure 3:	Top view of the 10-20 system for electrodes placement. ....	8
Figure 4:	An example of ten seconds of an EEG signal. ....	9
Figure 5:	Examples of artifacts. ....	10
Figure 6:	Properties of an ERP waveform. ....	11
Figure 7:	Typical amplitudes, frequencies, latencies, and waveforms of well-known ERPs. ....	12
Figure 8:	P3 waveform averaging ....	14
Figure 10:	Generall BCI schema. ....	18
Figure 11:	BCI calculator UI. ....	19
Figure 12:	Example of UI for BCI spelling system based on SSVEP. ....	20
Figure 13:	SSVEP based speller. ....	20
Figure 14:	Dividing of the EEG/ERP signal into epochs. ....	21
Figure 15:	BCI calculator detection result. ....	23
Figure 16:	LDA for two-class problem. ....	26
Figure 17:	An example of spectral analysis of an EEG trace. ....	27
Figure 18:	DSTFT principle. ....	28
Figure 19:	Well-known wavelets. ....	29
Figure 20:	Dilatation of the wavelet (Mexican hat). ....	30
Figure 21:	Translation of the wavelet (Mexican hat). ....	30
Figure 22:	Principle of discrete correlation. ....	31
Figure 23:	Scalogram. ....	31
Figure 24:	Haar wavelet. ....	31
Figure 25:	Principle of discrete wavelet transform. ....	32
Figure 26:	P3 component in the scalogram (CWT, Mexican hat). ....	34
Figure 27:	P3 component in the scalogram (DWT, Haar). ....	34
Figure 28:	Example of three iterations of MP algorithm ....	36
Figure 29:	The cross-component in Wigner-Ville transform. ....	37
Figure 30:	Input signal with P3 waveform. ....	38
Figure 31:	The atom which best approxiamtes the P3 waveform. ....	38
Figure 32:	Wigner-Ville transform of 20 Gabor atoms. ....	39

Figure 33:	The favorable decomposition of an input signal which contains the P3 waveform by the MP algorithm. ....	40
Figure 34:	MP iproved detection: the input signal.....	41
Figure 35:	MP iproved detection: reconstruction of the input signal.....	41
Figure 36:	MP iproved detection: ERP waveform model. ....	41
Figure 37:	EMD example .....	43
Figure 38:	Hilbert spectrum visualization. ....	45
Figure 39:	The mathematical model of a formal neuron.....	46
Figure 40:	The unsupervised learning process and ERP waveforms classification process of a general ANN.....	48
Figure 41:	Transformation of Gabor atom into feature vector. ....	49
Figure 42:	AT2 architecture. ....	50
Figure 43:	ART2P3 waveform cluster .....	52
Figure 44:	Schema of two-dimensional SOM architecture. ....	53
Figure 45:	Mask for visualization of weights similarity between neurons. ....	54
Figure 46:	Neuron weights similarity in a two-dimensional map. ....	55
Figure 47:	Mask which defines neighboring neurons during the first pass of the CCL algorithm. ....	55
Figure 48:	Neuron weights similarity in a two-dimensional map with manually highlighted clusters.....	56
Figure 49:	10-20 electrodes placement system from all views.....	69

## List of abbreviations

ANN	Artificial neural network
ANOVA	Analysis of variance
ART	Adaptive resonance theory
BCI	Brain-computer interface
CCL	Connected-component labeling
CCT	Cauchy convergence test
CFT	Continuous Fourier transform
CNS	Central nervous system
CSTFT	Continuous short-time Fourier transform
CWT	Continuous wavelet transform
DFT	Discrete Fourier transform
DSTFT	Discrete short-time Fourier transform
DWT	Discrete wavelet transform
EEG	Electroencephalography
EMD	Empirical mode decomposition
EPSP	Excitatory postsynaptic potential
ERP	Event-related potential
FFT	Fast Fourier transform
fMRI	Functional magnetic resonance imaging
FT	Fourier transform
HCI	Human-computer interface
HHT	Hilbert-Huang transform
HSA	Hilbert spectral analysis
ICA	Independent component analysis
IMF	Intrinsic mode functions
IPSP	Inhibitory postsynaptic potential
LBDWT	Lifting-based discrete wavelet transform
LDA	Linear discriminant analysis
MEG	Magnetoencephalography
MMN	Mismatch negativity
MP	Matching pursuit
PCA	Principal component analysis

PET	Positron emission tomography
SD	Standard deviation
SNR	Signal-to-noise ratio
SOM	Self-organizing map
SPECT	Single-photon emission computed tomography
SSVEP	Steady state visually evoked potentials
STFT	Short-time Fourier transform
SVM	Support vector machines
UI	User interface
WT	Wavelet transform

## Appendix A

<b>Brain activity monitoring method</b>	<b>Advantages</b>	<b>Disadvantages</b>
Non-invasive EEG	Cheap; easy to use; high time resolution	Low spatial resolution; low signal to noise ratio
Invasive EEG	Good spatial resolution In comparison with non-invasive EEG; high signal to noise ratio	Invasive
Magnetoencephalography (MEG)	Spatial resolution up to 3 mm; good time resolution; non-invasive method	Extremely expensive; the measurement is difficult; magnetically shielded room is necessary
Positron emission tomography (PET)	Better spatial resolution than SPECT; ability to detect which receptors are activated in the brain	Measured subject is exposed to radiation; it does not monitor an electrophysiological phenomenon - it monitors metabolism of oxygen and sugar
Single-photon emission computed tomography (SPECT)	In comparison with PET, it does not require a cyclotron	Measured subject is exposed to radiation; it monitors the blood circulation in the brain - not an electrophysiological phenomenon
Functional magnetic resonance imaging (fMRI)	High spatial resolution; non-invasive method	Extremely expensive; time resolution cca 1 second.

Table 2: Comparison of methods for monitoring of the brain activity. [66; 67; 68]

## Appendix B

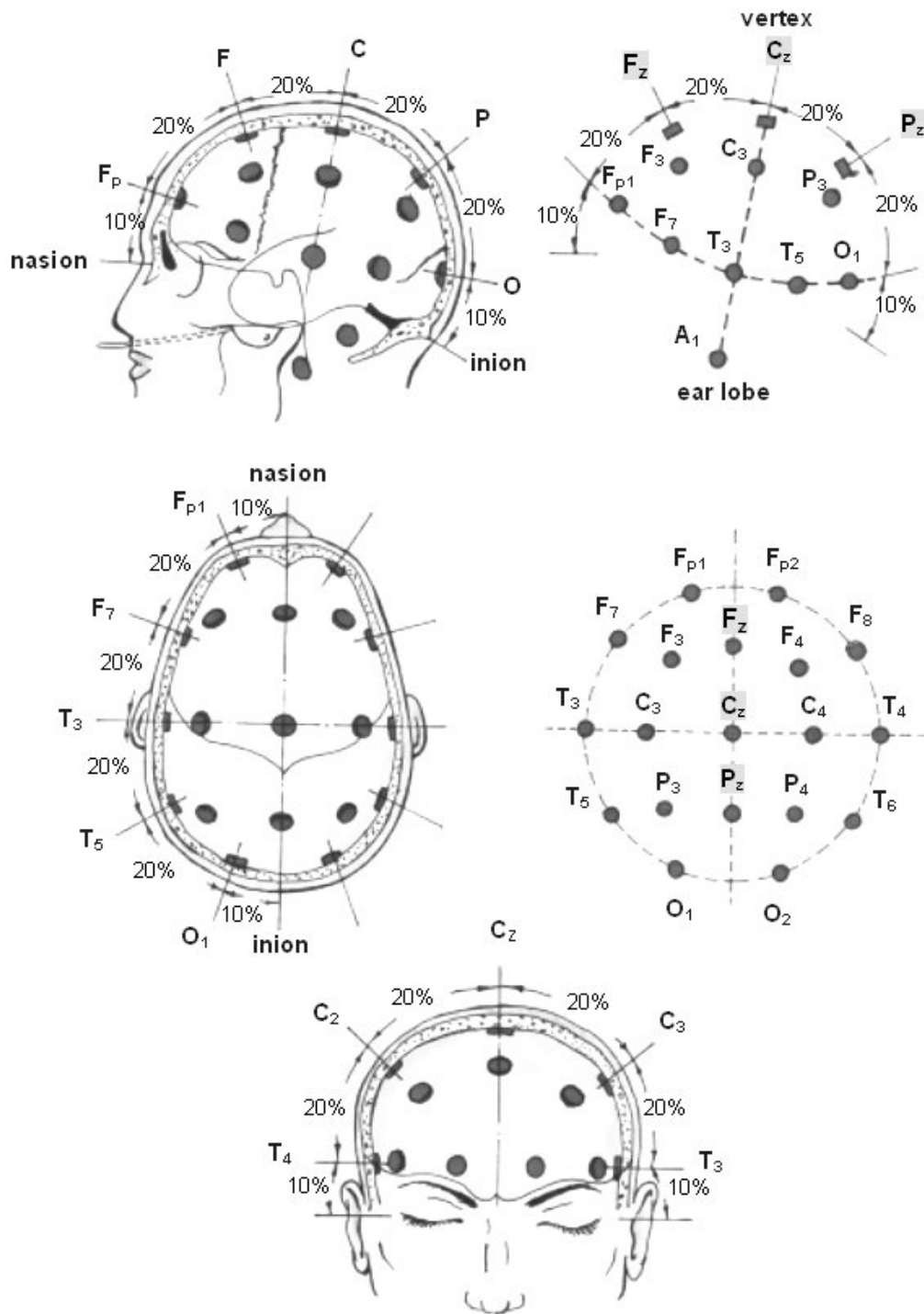


Figure 49: 10-20 electrodes placement system from all views.