



TEKNILLINEN KORKEAKOULU

**Tommi Nykopp**

**Statistical Modelling Issues for The  
Adaptive Brain Interface**

Diplomi-insinöörin tutkintoa varten tarkastettavaksi jätetty diplomityö

Työn valvoja: Akatemia Professori Kimmo Kaski

Työn ohjaaja: TkT. Jukka Heikkonen

Espoo 9.6.2001

<b>Tekijä:</b>	Tommi Nykopp	
<b>Otsikko:</b>	Adaptiivisen Aivokäyttöliittymän Tilastollisesta Maliinnuksesta	
<b>Päivämäärä:</b>	9.6.2001	<b>Sivumäärä:</b> 113
<b>Osasto:</b>	Sähkö- ja Tietoliikennetekniikan osasto	
<b>Professori:</b>	S-114, Laskennallinen tekniikka	
<b>Valvoja:</b>	Akatemia Professori Kimmo Kaski	
<b>Ohjaaja:</b>	Akatemian vanhempi tutkija, TkT Jukka Heikonen	
<p>Tässä työssä tutkittiin aivokäyttöliittymiin liittyvää EEG signaalin esikäsittelyä ja tilastollista mallinnusta. Aivokäyttöliittymä on laite, jonka avulla eri sovelluksia voidaan hallita pelkästään tietoisella ajatusten hallinnalla. Mitään kehon muuta normaalia informaatioväylää, kuten lihas -tai hermotoimintaa, ei tarvita.</p> <p>Työssä keskityttiin aivokäyttöliittymien kaikkiin kolmeen tilastollisen mallinnuksen osa-alueeseen: Esikäsittelyyn, piirreirroitukseen ja luokitteluun. Eri esikäsittelymenetelmiä olivat spatiaalinen Laplacian suodatus, kaistanpäästösuodatus ja aallokehajotelmaan poh-jautuva kohinanpoisto. Piirreirroituksessa tutkittiin eri tapoja estimoida EEG signaalin tehospektriä. Käytettyjä menetelmiä olivat Fourier -ja aallokeestimaatit, sekä autoregressiomallin siirtofunktioon perustuva estimaatti. Tilastollisista luokittelijoista tutkittiin monikerrosperceptron -ja radiaalikantafunktioverkkoja. Suorituskyvyn mittareina käytettiin mallin tuottamaa kanavakapasiteettia ja väärin luokitusten määrää.</p> <p>Esikäsittelymenetelmistä spatiaalinen Laplacian suodatus osoittautui välttämättömäksi hyvän luokittelutuloksen saamiseksi. Piirreirrotusmenetelmistä aalloke -ja autoregressiomalliin perustuvat estimaatit olivat suhteellisesti parempia. Monikerrosperceptron-verkoissa luokitustulokset olivat korkeammat, mutta radiaalikantafunktioverkoissa esiintyi vähemmän väärää luokituksia.</p>		
<b>Avainsanat:</b> aivokäyttöliittymä, aivoaktiviteetti, spektriestimaatti, tilastollinen mallinnus.		

<b>Author:</b>	Tommi Nykopp	
<b>Title:</b>	Statistical Modelling Issues for The Adaptive Brain Interface	
<b>Date:</b>	9.6.2001	<b>Number of Pages:</b> 113
<b>Department:</b>	Department of Electrical and Communications Engineering	
<b>Professorship:</b>	S-114, Computational Engineering	
<b>Supervisor:</b>	Academy Professor Kimmo Kaski	
<b>Instructor:</b>	Academy Fellow, Dr. Tech Jukka Heikkonen	
<p>In this work methods for Electroencephalography (EEG) signal preprocessing and statistical modelling related to the brain computer interfaces were studied. Brain computer interface is a device for controlling various applications by user's conscious control of thoughts. No use peripheral nerves or muscles is required.</p> <p>All the main areas of statistical modelling related to the brain computer interfaces were studied: preprocessing, feature extraction and classification. The used preprocessing methods were spatial Laplacian filtering, bandpass filtering and wavelet denoising. In the feature extraction different ways to form a spectrum estimate were studied. Studied method were Fourier -and wavelet estimates and the estimate based on transfer function of an autoregressive (AR) model. For the classification two different statistical models were used. The first was multi-layer perceptron network and the another radial basis function network. The performance was measured by achieved channel capacity and false positive rate.</p> <p>From the preprocessing methods the spatial Laplacian filtering was found to be essential. The wavelet and AR-model based features were a bit more effective than the Fourier features. The multi-layer perceptron networks had better classification results but the radial basis function networks had fewer false positives.</p>		
<b>Keywords:</b> Brain computer interface, brain activity, spectrum estimation, statistical modelling		

*I think, therefore I am.*

- René Descartes

# Foreword

This Master's Thesis has been done in Laboratory of Computational Engineering in Helsinki University of Technology as a part of "*Adaptive Brain Interfaces*" project funded by the ESPRIT Programme of the European Commission. The supervisor of this work was Academy Professor Kimmo Kaski and instructor Academy Senior Researcher Jukka Heikkonen.

Here I would like to thank Prof. Kimmo Kaski for opportunity to work in this fascinating project. Also I would like to thank all the partners of this project with whom I have had a privilege to work with. Especially I thank Prof. José del R. Millán and M.Sc Josep Mouriño for all the advice and the companionship. This laboratory has been an excellent place to work in so I thank all the personnel here, especially Dr.Tech Jukka Heikkonen for being my excellent instructor and there is no way I could ever thank M.Sc Markus Varsta enough.

Otaniemi 25th February 2002

Tommi Nykopp

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	The Brain . . . . .	3
1.1.1	The structure of the brain . . . . .	3
1.1.2	The Cortex . . . . .	4
1.1.3	The Lobes of the Cortex . . . . .	6
1.1.4	The Basal Ganglia . . . . .	7
1.1.5	The Thalamus and Hypothalamus . . . . .	7
1.2	On the Generation of the EEG signal . . . . .	8
1.2.1	Communication between the brain cells (neurons) . . . . .	8
1.2.2	Frequency bands of the EEG . . . . .	11
1.2.3	Evoked Responses from the User . . . . .	13
1.2.4	Measuring the EEG signal . . . . .	15
1.3	Brain Computer Interface (BCI) . . . . .	21
1.3.1	What Is BCI . . . . .	21
1.3.2	System Components . . . . .	21

1.3.3	Training . . . . .	23
1.3.4	Adaptive Brain Interface . . . . .	24
1.3.5	Applications . . . . .	27
1.4	Other BCI systems . . . . .	28
1.4.1	VEP Detection . . . . .	28
1.4.2	P300 Detection . . . . .	30
1.4.3	Slow Cortical Potentials (SCP) . . . . .	31
1.4.4	EEG $\mu$ -rhythm Conditioning . . . . .	32
1.4.5	EEG Pattern Recognition Approach . . . . .	35
<b>2</b>	<b>Statistical Modelling</b>	<b>38</b>
2.1	Used Data . . . . .	39
2.1.1	Baseline removing . . . . .	40
2.1.2	Removing of the samples near class transition . . . . .	40
2.2	Preprocessing . . . . .	41
2.2.1	Laplacian filtering . . . . .	41
2.2.2	Bandpass filtering . . . . .	42
2.2.3	Wavelet denoising based on NML criterion . . . . .	42
2.3	Feature extraction . . . . .	49
2.3.1	Fourier spectral features . . . . .	49
2.3.2	Autoregressive spectral features . . . . .	52
2.3.3	Wavelet spectral features . . . . .	53

2.3.4	Calculating the Features . . . . .	53
2.4	Classifiers . . . . .	55
2.4.1	Multi Layer Perceptron (MLP) committee with Early Stopping regularisation . . . . .	56
2.4.2	Radial Basis Function (RBF) nets . . . . .	58
<b>3</b>	<b>Experiments</b>	<b>61</b>
3.1	Definitions . . . . .	61
3.2	Training and Testing Procedure . . . . .	62
3.3	Experiment 1. . . . .	63
3.4	Experiment 2. . . . .	63
3.5	Experiment 3. . . . .	63
3.6	Experiment 4. . . . .	64
<b>4</b>	<b>Results</b>	<b>65</b>
4.1	Experiment 1. . . . .	65
4.2	Experiment 2. . . . .	67
4.3	Experiment 3. . . . .	68
4.4	Experiment 4. . . . .	71
4.5	Summary of the Results . . . . .	72
<b>A</b>	<b>Weight Optimization Algorithms</b>	<b>79</b>
A.1	Resilient Propagation (RProp) . . . . .	79

A.2	Scaled Conjugate Gradient (SCG)	80
A.3	Forward Selection	83
<b>B</b>	<b>Channel Capacity</b>	<b>86</b>
B.1	Channel Capacity Performance Measure	86
B.1.1	Definition of channel capacity	87
B.1.2	Channel Capacity for ABI	88

# List of Figures

1.1	Brain 'coordinates'. The principal directions and areas how a brain is described. Source: <a href="http://www.biomag.helsinki.fi/braincourse/L1.html">http://www.biomag.helsinki.fi/braincourse/L1.html</a> . . . . .	4
1.2	A coronal section of the human brain on which the sensory and motor cortices are laid out on the post- and precentral strip, respectively. Note that the somatosensory sectioning is located behind the motor sectioning as shown here. [Hug95] . . . . .	5
1.3	The four cortical lobes of the brain seen from the side ( <i>lateral</i> ) the midline ( <i>medial</i> ) and behind and above the head ( <i>dorsal</i> ). [Hug95] . . . . .	6
1.4	Midline sectioning of the brain, including the brainstem . [Hug95] . . . . .	8
1.5	SCP development for linguistic and mental visualisation tasks. [Nie99c]	15
1.6	On the left: the standard <i>10-20</i> electrode configuration. On the right: the <i>10-20</i> system augmented to 75 electrode configuration. [Nie99d] . . . . .	16
1.7	Above: Bipolar linkage over two electrodes Below: Bipolar linkage over there electrodes. [Nie99d] . . . . .	17
1.8	The key components of the ABI . . . . .	24
1.9	Virtual keyboard application for producing text . . . . .	27
1.10	Robot application for ABI . . . . .	28
1.11	Classical Pacman game being played with ABI . . . . .	29
2.1	The problem of high variances of classes 1 (blue), 2 (red) and 3 (green) . . . . .	39

2.2	FIR lowpass filter of order 8 . . . . .	42
2.3	FIR highpass filter of order 20 . . . . .	43
2.4	Bandpass filtered 0.5s segment of EEG . . . . .	44
2.5	a) The Fourier transform is localized only in time. b) The wavelet transform is localized both in time and frequency. . . . .	45
2.6	Scaling of a wavelet basis function . . . . .	46
2.7	Shifting of the basis function with parameter $a = k$ . . . . .	47
2.8	The effect of the filter order to denoising . . . . .	48
2.9	Effect of the windowing to a raw signal . . . . .	50
2.10	Power spectras extracted with two different method . . . . .	52
4.1	Preprocessing results for the MLP classifier with $\rho = 90\%$ . . . . .	66
4.2	Preprocessing performances for the MLP classifier with $\rho = 90\%$ . . . . .	67
4.3	Preprocessing results for the RBF classifier with $\rho = 50\%$ . . . . .	69
4.4	Preprocessing performances for the RBF classifier with $\rho = 50\%$ . . . . .	70
4.5	Daily results of the different Fourier features with $\rho = 90\%$ . . . . .	71
4.6	Daily results for the MLP classifier with the different features and $\rho = 90\%$	72
4.7	Daily performances for the MLP classifier with the different features and $\rho = 90\%$ . . . . .	75
4.8	Daily results for the wavelet features with $\rho = 90\%$ . . . . .	76
4.9	Daily results for the wavelet features with $\rho = 90\%$ . . . . .	76
4.10	Daily results for the RBF classifier with the different features and $\rho = 50\%$	77

4.11 Daily performance for the RBF classifier with the different features and  $\rho = 50\%$  . . . . . 77

4.12 Daily results for the weight optimization algorithms . . . . . 78

A.1 Problem of the gradient descent: local gradients  $-\nabla E$  do not point towards error minimum, instead they oscillate across “the error valley” . . . 80

A.2 Conjugate directions:  $w_2$  is minimum point on the line starting at  $w_1$  to direction  $d_1$ . If the component of the gradient of  $w_2$  to direction  $d_2$  parallel to previous line remains zero, then the direction  $d_2$  is said to be *conjugate* to direction  $d_1$  . . . . . 81

A.3 Problem of the gradient descent: local gradients  $-\nabla E$  do not point towards error minimum, instead they oscillate across “the error valley” . . . 82

B.1 The components of the total entropy of the system . . . . . 88

# List of Tables

1.1	The most important frequency bands and $\mu$ -rhythm of brain activity and their details. The $\mu$ -rhythm is included to this list even though it is not actually a band but an important rhythm in BCI research centered between 9 - 11 Hz . . . . .	11
2.1	Summary of the recorded EEG data . . . . .	40
2.2	Used mental tasks . . . . .	40
3.1	An example of a confusion matrix . . . . .	61
4.1	Left: Results from Unprocessed MLP. Right: Results from Laplacian filtered MLP. $\rho = 90\%$ . . . . .	68
4.2	Daily performances of the different Fourier features with $\rho = 90\%$ . . . . .	68
4.3	Left: Results from MLP+AR features with $\rho = 0$ . Right: Results from MLP+AR with $\rho = 90$ . . . . .	73
4.4	Left: Results from RBF+AR features with $\rho = 0$ . Right: Results from RBF+AR with $\rho = 50$ . . . . .	73
4.5	Daily performances of the weight optimization algorithms with $\rho = 90\%$ . . . . .	74
4.6	Collected results for the day 1. . . . .	74
4.7	Collected results for the day 2. . . . .	75

4.8	Collected results for the day 3. . . . .	78
-----	--	----

# Nomenclature

## Abbreviations

AR	AutoRegressive model
ABI	Adaptive Brain Interface
BCI	Brain Computer Interface
CC	Channel Capacity
ECG	Electrocardiography, Electric activity in the heart
EEG	Electroencephalography, Electric brain activity
EMG	Electromyography, Electric activity in the muscles
EOG	Electrooculography, Electric activity in the eyes
EP	Evoked Potentials
ERD/ERS	Event Related Synchronizations/Desynchronizations
ERP	Event Related Potentials
ES	Early Stopping -regulation
F	Fourier
FIR	Finite Impulse Response
FPS	False Positive Rate
GCV	Generalized Cross Validation
FS	Forward Selection model order selection method
Lap	Laplacian Filtering
ML	Maximum Likelihood
MLP	Multi Layer Perceptron
NML	Normalized Maximum Likelihood
MSE	Mean Square Error
NN	artificial Neural Network
P300	Event related Potential occurring 300 ms after stimulus
RBF	Radial Basis Function

Rprop	Resilient Propagation
SCG	Scaled Conjugate Gradient
SCP	Slow Cortical Potential
SSE	Sum of Square Errors
VEP	Visually Evoked Potential
Wav	Wavelet

# Symbols

$\alpha$	alpha band EEG frequencies; control parameter in SCG weight optimization
$\beta$	beta band EEG frequencies
$\delta$	delta band EEG frequencies
$\Delta$	constant weight step in Rprop weight optimization
$\Theta$	theta band EEG frequencies
$\theta$	parameter of a model
$\hat{\theta}$	maximum likelihood estimate of a parameter
$\eta$	increase/decrease factor in Rprop weight optimization
$\lambda$	regularization parameter; scaling coefficient in SCG weight optimization
$\mu$	Rolandic motor band EEG frequencies; mean of a normal distribution
$\Psi$	continuous wavelet basis function
$\psi$	discrete wavelet basis function
$\Sigma$	covariance of a normal distribution
$\sigma^2$	standard deviation of a normal distribution
$a$	wavelet shifting parameter; coefficient of an autoregressive model
$b$	wavelet scaling parameter
$c$	number of classes or outputs; wavelet coefficient
$C_k$	class $k$
$d$	number of inputs
$E$	error function
$g(\cdot)$	activation function
$\mathbf{H}$	Hessian matrix
$H(z)$	transfer function on a z-plane
$h$	basis function
$\mathbf{h}$	design matrix
$I$	NML -criterion score
$i$	input label
$j$	hidden unit label
$K$	wavelet coefficients
$k$	output unit label
$M$	number of hidden units

$m$	order of a model
$n$	pattern label
$N$	number of patterns
$N(\mu, \sigma^2)$	normal distribution of mean $\mu$ and standard deviation $\sigma^2$
$P(\cdot)$	probability
$\mathbf{P}$	projection matrix
$p(\cdot)$	probability density function
$\mathbf{R}$	noise variance
$\rho$	classification probability threshold
$S$	cost function
$t$	target value
$\tau$	time step in iterative algorithms
$W$	number of weights and biases in a network; wavelet transformation matrix
$\omega$	weight
$x$	network input variable
$y$	network output variable
$z$	activation of hidden unit

# Chapter 1

## Introduction

This thesis describes the principles of a communication system called *Brain Computer Interface (BCI)*. With this system user can control applications by using his/hers brain activity alone, no peripheral muscles or nerves are required. The brain activity can be used for communication by classifying the activity to different tasks, which correspond to the functions in used application e.g. pressing a key or moving a mouse.

The user concentrates to different mental tasks, which activate different functional areas of the brain. This activity is measured as the *Electroencephalography (EEG)*, and from its certain features, usually the power spectrum of the EEG, are extracted.

These features are then classified by a statistical model, usually an artificial neural network. The model is trained in co-operation between the user and the model, where both adapt to each other, the user by using *biofeedback* and the model by using the features received from the user.

In this work a BCI called *Adaptive Brain Interface* is described. Also the classification of the mental tasks is studied. In the classification procedure the measured EEG is first sampled and segmented, then it is preprocessed by filtering and denoising. Then the features are extracted for the classification, which is done by a statistical model.

In the first chapter some basic principles of the brain and its functionality is described. Also the EEG measurement is described and an introduction to some of the current BCI systems is provided, giving more close look to the ABI system.

The second chapter presents more in depth the statistical modelling required in the ABI

system and describes the different preprocessing, feature extraction and classification methods used in this work.

In the third chapter the experiment environment is described, including the used data, mandatory preprocessing and experiment setup. The results with analysis are provided in the fourth chapter.

The appendixes *A* and *B* concern weight optimization of the classifiers and the channel capacity performance measure, respectively.

## 1.1 The Brain

All of it happens in the brain. The brain is undoubtedly the most complex organ found among the carbon based lifeforms. So complex it is that we have only vague information about how it works. This section tries to give an overview of the structure and the functions of the brain and is based on the material from [Ilm00], [Hug95] and [Nie99e].

### 1.1.1 The structure of the brain

The brain is part of the *central nervous system*, which consists of *large brain*, *little brain*, *brainstem* and *spinal cord*. Brainstem connects the brain to the spinal cord. If a fully developed brain is view at the bottom, the first part to be seen is the *myelencephalon*, which means the *spinal brain*. Above the myelencephalon is the *metencephalon*, which means the *across brain* and consists of Cerebellum and fourth ventricle. These mentioned parts can be considered as *hind brain*. The second part, *mid brain* consists of *mesencephalon*, where are the *tectum*, *tegmentum* and *cerebral aqueduct*. The third part, the *fore brain*, is divided into two parts, *diencephalon* and *telencephalon*. The brain floats in a substance called *cerebrospinal fluid*, which purpose is provide a stable platform for the brain to operate and keep the brain in optimal working conditions. The brain and the cerebrospinal fluid are packed into *dura matter*, which is a 'bag' that holds them. And these are then protected by a skull, covered with skin (*scalp*) and hair.

From a functional perspective, the brain is divided into three parts. The first is the so called the *large brain*, which is also known as *fore brain* or *cerebrum*. This first part controls higher mental function such as languages and abstract reasoning and consists of *diencephalon* and *telencephalon*. The second part is *brainstem*, where visual and auditional functions happen. The brainstem is located in *mesencephalon*. The third part is the *cerebellum* or the *little brain* and it handles the motor control and movement. Next a closer look of some of the key areas in the brain is taken.

In figure (1.1) the 'coordinate system' of the brain in described. In the following text the location of the different parts of the brain is usually given in terms of figure (1.1).

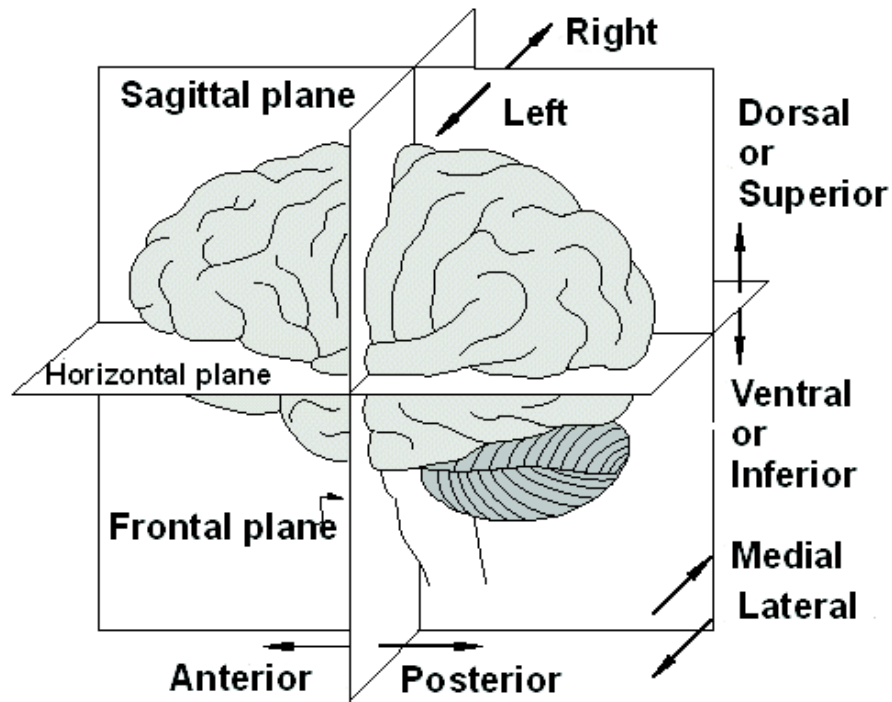


Figure 1.1: Brain 'coordinates'. The principal directions and areas how a brain is described. Source: <http://www.biomag.helsinki.fi/braincourse/L1.html>

## 1.1.2 The Cortex

The dominant part of the cerebrum, the cortex is the convoluted outer part of the brain. It consists of  $10^{10}$  to  $10^{12}$  neurons arranged into six different layers. The cortex is rather thin, only 2-3mm, but its total area exceed the area of the skull. This is due the different shapes of the cortex surface, such as the *fissures*, which are foldings that divide the brain the into the two hemispheres and to frontal and temporal lobes. Another type of folding is the *sulcus*, which e.g. divides the frontal and parietal lobes.

### Cerebral Hemispheres

The left and right hemispheres of the brain are the major sites for higher cognitive functions. The left hemisphere serves for functions related for language and verbal materials,

and in to some extent to positive emotions, whereas the right side to visuo-spatial function and negative emotions. The two cerebral hemispheres are connected through the *corpus callosum*, which is accumulation of million of nerve fibers, which run across the left end the right hemisphere and it has a shape of a horseshoe. The hemispheres are also connected by *anterior and posterior commissures*, which are located anterior and posterior to the third verticle.

### The Somamotor Cortex and the Somatosensory Cortex

A major functional division between cortical area is the division between the motor and the sensory cortices. In this point it should also some other parts of the brain, such as basal ganglia, are involved in motor functions. Figure (1.2) is a guiding picture showing how some of the bodily function are located in the cortex. The key rule in functional

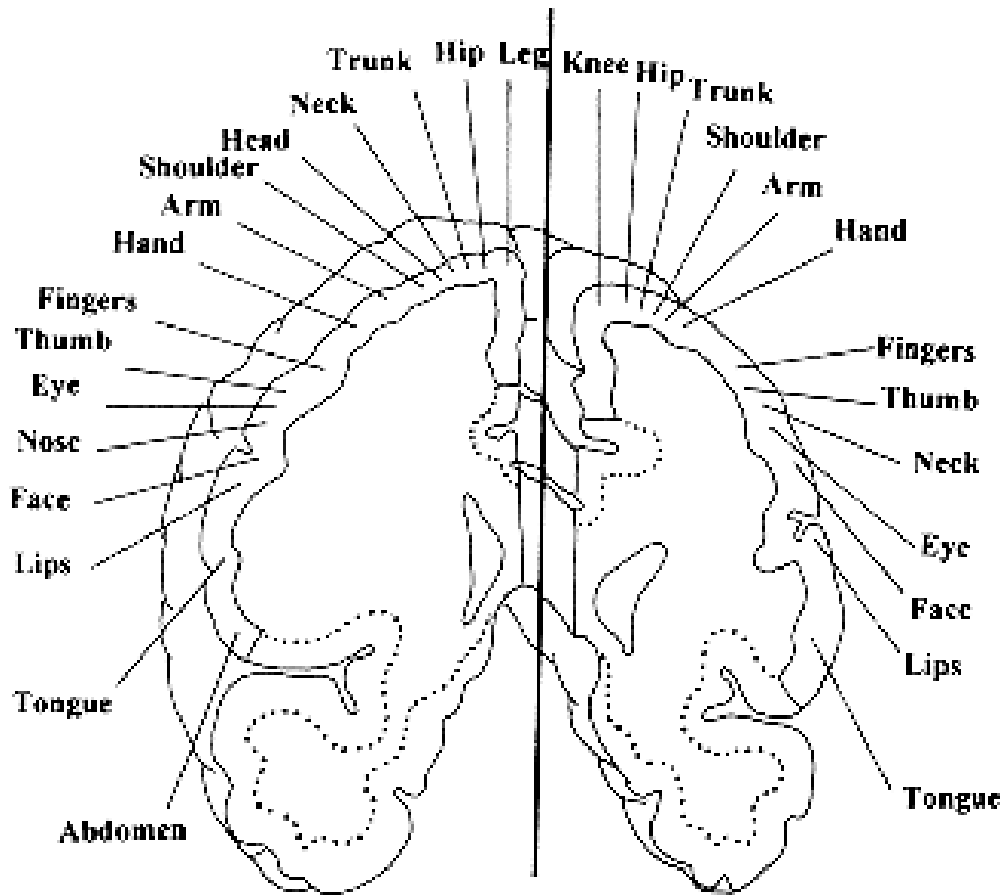


Figure 1.2: A coronal section of the human brain on which the sensory and motor cortices are laid out on the post- and precentral strip, respectively. Note that the somatosensory sectioning is located behind the motor sectioning as shown here. [Hug95]

localization is that the more finely graded motor control is needed, the larger is the area

occupied by that part on the motor cortex, e.g. thumbs and lips occupy significantly larger area than the torso and the legs. Somatosensory cortex is laid similarly like the motor cortex as shown also in figure (1.2).

An important although well known organizational principle of the brain and the nervous system, is the *contralateral crossing*, which means that left side is controlled by the right and vice versa. For example the left hand is controlled by motor cortex in the right side of the brain.

### 1.1.3 The Lobes of the Cortex

There are four lobes of the cortical surface, the *frontal*, *temporal*, *parietal* and *occipital* each having two halves over the longitudinal fissure, as shown in figure (1.3).

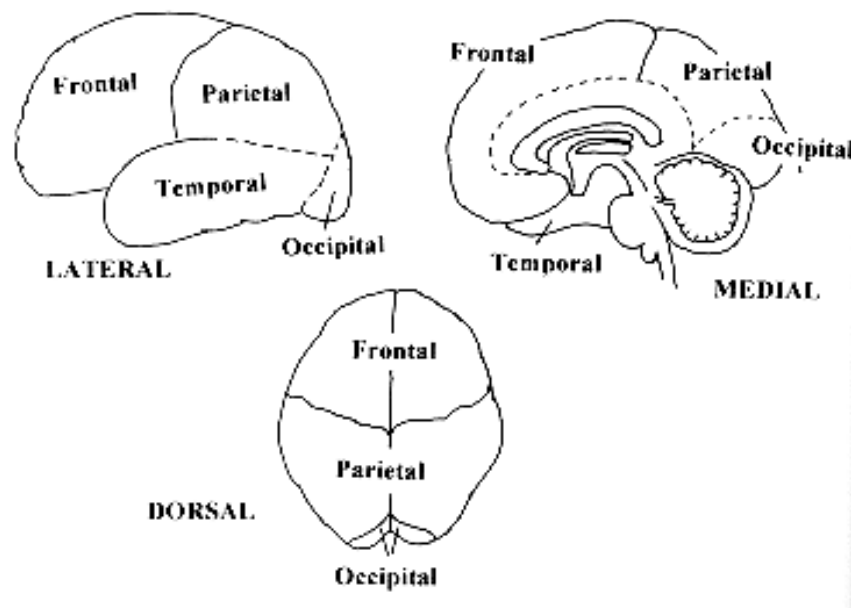


Figure 1.3: The four cortical lobes of the brain seen from the side (*lateral*) the midline (*medial*) and behind and above the head (*dorsal*). [Hug95]

The frontal lobe controls complex cognitive function such as attention and language and programmes and executes motor patterns. Damage to the frontal lobe can lead to degenerative diseases, such as Parkinson's or Alzheimer's disease or severe psychosis, such as schizophrenia.

In the temporal lobe the main functions are the auditory functions and other functions that

are performed here include storage of visual memory, processing of spoken and heard language. Medial in the temporal lobe is a structure called *hippocampus*, which is related to memory, particularly to the consolidation of memory. Much of the past years research in learning and memory is concentrated to hippocampus and its surrounding areas.

The parietal cortex is the first of all involved in sensing touch and kinesthetics. It is also involved in integrating information from different senses, e.g. when we see and hear a car, parietal cortex is involved in forming a unitary phenomenon, a *car*.

The occipital lobe contains the major visual area of the brain. This is the place where the visual information is directed from the eye. It is also involved in another visual functions, such as recognition.

#### **1.1.4 The Basal Ganglia**

The basal ganglia help to control the body movements by integrating sensory and motor information from other areas of the brain. Dysfunctions in dopamine activity in the basal ganglia causes Parkinson's disease, which is characterized by tremor and rigidity in the limbs.

#### **1.1.5 The Thalamus and Hypothalamus**

The division of the thalamus into different nuclei and functional systems is shown in figure (1.4). The thalamus is a kind of a *relay station*, which integrates and passes on somatomotory and somatosensory information.

The hypothalamus controls homeostatic bodily function, such as temperature, sex, thirst, hunger, etc. by controlling levels of certain hormones.

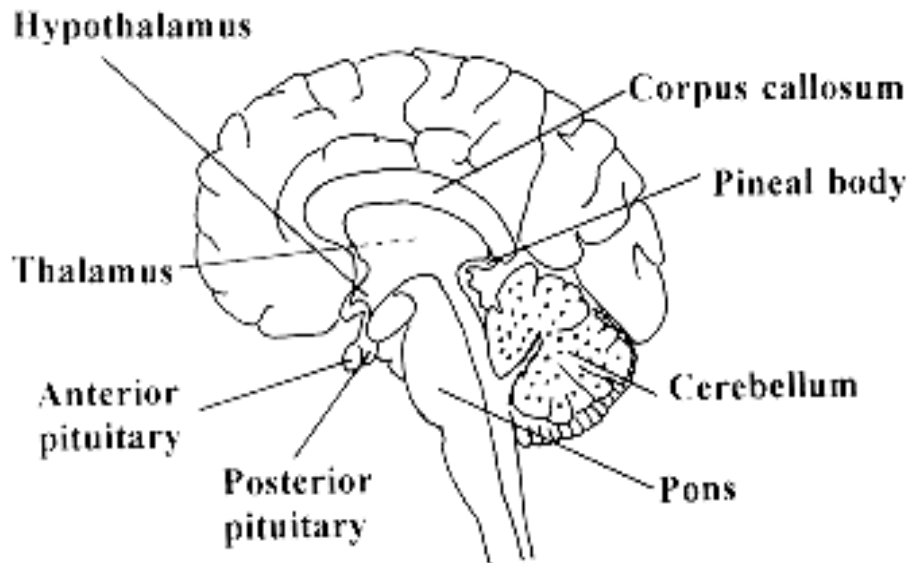


Figure 1.4: Midline sectioning of the brain, including the brainstem . [Hug95]

## 1.2 On the Generation of the EEG signal

In this section some basic principles about *Electroencephalography* or EEG for short and its measurement are described. As the neurons in our brain communicate with each other by firing electrical impulses, this creates an electric field which travel though the cortex, the dura, the skull and the scalp (see section 1.1.1). The EEG is measured from the surface of the scalp by measuring potential difference between the actual measuring electrode and a reference electrore. First, the generation of the electric potential is described, then the actual measurement techniques and issues influencing it.

### 1.2.1 Communication between the brain cells (neurons)

The information between the braincells is relayed from *dendrites*, which is the input channels to cell, to *axon*, which is the output channel. The cell wall is permeable for sodium and potassium ions, and this permeability is function of the electric potential of the cell wall. These ion currents make possible that an unattenuated electric field, *action potential*, can propagate through the cells.

In order to relay information, the ion concentrations outside and inside the cell must be different: on the outside should be lots of potassium and inside the cell should be lots of sodium ion. The amount of these ions in the cell is controlled by the *Na-K pumps* which

pump the unnecessary ions out or in correspondingly. Pumping is done to 'upstream', i.e. from lower concentration to higher concentration.

The potential on the inside of the cell is about  $70mV$  smaller than the potential on the outside. When the potential is decreasing, it is called *depolarization* and when increasing, *hyperpolarization*. When depolarization is big enough, a certain threshold is reached, and more sodium ions flood to cell depolarizing it further. Then the cell will 'discharge' and send an action potential, which floods potassium ions into the cell and turns its electric charge to positive. As the action potential propagates through the cells, its amplitude remains constant. This phenomenon travels from cell to cell like is the 'domino-effect'. No energy or material propagates in this process, just the information. If the depolarization does not reach the threshold, it will be only a local change in potential. The action potential can propagate as fast as  $100m/s$ .

The information processing in a network of braincells is based on connections between the braincells. The actual connection is made by *synapses*, which are located at surfaces of the dendrites and axons. When comparing to the artificial neural networks used in statistical modelling, synapses are bit similar to *weights* of a neuron. The rate in which a braincell relays information depends on weighted sum of the input signals. The weight of a synapse depends its distance from *soma*, which is the center of a cell.

The cells affect each other either *excitatory* i.e. increasing the action, or *inhibitory* i.e. inhibiting the action between the cells. On the surface of a cell there is usually large amount synapses of both types. When signal arrives to a synapse, it releases a chemical *transmitter substance* to a gap in the synapse, from where the transmitter diffuses to the cell wall of the postsynaptic cell. In an excitatory synapse the transmitter causes to positive ions to flood into the postsynaptic cell to depolarize the cell wall. In an inhibitory synapse, the transmitter tries to keep the potential of the cell wall below the threshold.

A neural network operates by individual cells as they discharge action potentials corresponding to synaptic information they receive. A cell sums all the inputs that it gets and if the difference between the excitatory and inhibitory is big enough, i.e. the threshold is reached, it sends its own signal forward. The input signal can also change neurons growth, metabolism and the weight of a synaptic connections.

The fundamental assumption behind the EEG signal is that it reflects the dynamics of electrical activity in populations of neurons. The crucial property of such populations is that they can work in synchrony. In order to be able to work in synchrony, connections between the neurons must be formed to build up a network of neurons. The terminology

to describe such networks has been developed by Freeman [Fre75]. A basic unit of such a network is referred as *KI* sets, which are populations of neurons with mutual inhibitory (*KIi*) or excitatory (*KIe*) interaction. An interacting pair of *KIi* and *KIe* set form *KII* set, and further *KIII* set is formed by two interacting *KII* sets. A group of *KIII* sets usually occupy an area of a few square millimeters in the cortical surface or a nuclear volume of a few cubic millimeters in the brainstem and a spinal cord. A neural mass usually consists between  $10^4$  and  $10^7$  neurons.

These are the basic sets that generate the oscillatory phenomena measured as EEG signal. The potential fluctuation that we measure as EEG signal is generated by sum of the postsynaptic potentials in the cortex. Since a single postsynaptic potential generates a potential of magnitude  $10^{-14}V$ , thousands of these synapses must activate synchronously in order to be detectable with EEG.

The EEG is supposed to be generated by oscillations between the cortex and the thalamus. These oscillations seems to be generated by physical properties of a neuron and by functions of the ionic channels in the cell walls of the thalamic cells. Research seems to point out that functional states in the cortex such as (*sleep, information processing or relaxation*) are caused by changes in thalamic activity.

The process which generates the EEG is very complex due to the large amount of the independent neurons. Therefore the research for the models of the EEG signal generation is a kind of dialogue of theory and experimentation, which can be described as two part process. First the theory makes assumptions about the process which is then tested by experimenting, i.e. testing different inputs and studying their outputs or changing some of the properties of the constituting elements. The hypotheses are formulated concerning new elementary properties, relationships and overall behaviour. These new hypotheses then predict new results, raise new questions and suggest new experiments to validate given hypotheses.

This dialogue has so far produced several theories and models about EEG signal generation and in the last years it seems to be that the following models have raised the most interest [Nie99a]

- The model describing the generation of the EEG of olfactory areas of the brain, proposed by Freeman in 1975
- The model of the alpha rhythm in the thalamus and cortex, proposed by Lopes da Silva et al. in 1974

- The series of models of the membrane and synaptic properties of thalamic cells and circuits responsible of the generation of 7 - 14 Hz spindle rhythmicity that occurs in the light stages of sleep, based on the experimental findings of Steriade et al. in 1993
- The model of the generation epilepticform transients proposed by Traub in 1982
- The model of gamma rhythms developed by Traub et al. in 1996 and realistic simulations of synapses in 1996

## 1.2.2 Frequency bands of the EEG

Some of the frequency bands found from the EEG are shown in table (1.2.2). Most of the brain research is concentrated in these channels and especially  $\alpha$  and  $\beta$  bands are important for BCI research. The reason why the bands do not follow the greek letter

Band	Frequency [Hz]	Amplitude [ $\mu V$ ]	Location
<b>Alpha</b> ( $\alpha$ )	8 - 12	10 - 150	Occipital/Parietal regions
$\mu$ -rhythm	9-11	varies	Precentral/Postcentral regions
<b>Beta</b> ( $\beta$ )	14 - 30	25	typically frontal regions
<b>Theta</b> ( $\theta$ )	4 - 7	varies	varies
<b>Delta</b> ( $\delta$ )	< 3	varies	varies

Table 1.1: The most important frequency bands and  $\mu$  -rhythm of brain activity and their details. The  $\mu$  -rhythm is included to this list even though it is not actually a band but an important rhythm in BCI research centered between 9 - 11 Hz

magnitudely (alpha is not the lowest band) is that this is the order in which they were discovered.

The EEG contains quite a wide spectrum of frequencies but it is not just an even mixture of things. EEG has organization and rhythmicity but only to certain level. Too much rhythmicity may indicate abnormality but chaotic and seemingly noisy signal may not.

The overall bandwidth of the EEG is about 0.1 Hz - 100 Hz but the practical limit is 0.3 and 70 Hz.

The EEG amplitude is measured peak to peak and its accurate determination is difficult. Therefore EEG amplitudes are usually noted as certain difference or even verbally, e.g. medium voltage or low to medium voltage.

## Alpha Rhythm

The official definition of the  $\alpha$  -rhythm is: *Rhythm at 8-13 Hz occurring during wakefulness over the posterior regions of the head, generally with higher voltage over occipital areas. Amplitude is variable but mostly below  $50\mu V$  in adults. Best seen with (the patients) eyes closed and under physical relaxation and relative mental inactivity. Blocked or attenuated by attention, especially visual and mental effort. (IFSECN, 1974)*

The mean  $\alpha$  -rhythm of adult male is  $10.2 \pm 0.9$  Hz and it decreased as person ages, most probably due to the degeneration of cerebral.  $\alpha$  -Rhythm is characterized by sinusoidal wave. Spatially the  $\alpha$  -rhythm is a manifestation the posterior half of the head and found over occipital, parietal and posterior lobes.

$\alpha$  -Rhythm is temporarily blocked by an eye opening (influx of light) and mental activities, which are usually less effective than eye opening. It has been proposed that  $\alpha$  -rhythm is a kind of a *standing by* -state of the brain, ready for switching to other activity bands according to activity (sloop, drowsiness, cognitive tasks).

## $\mu$ -Rhythm

The mu -Rhythm, sometimes called Rolandic  $\mu$  -rhythm, is in frequency and amplitude related to posterior  $\alpha$  -rhythm, but its topography and physiological meaning is quite different. The ' $\mu$ ' stands for motor and it is strongly related to movement functions of the motor cortex. This rhythm is very asymmetric, the negative side being very sharp and spiky and the positive being rounded. Most commonly detected at frequencies between 9-11 Hz and values of 8 Hz or below may indicate brain abnormality. This rhythm is detected over the precentral and postcentral region, using electrodes of C3 and C4 of the standard 10-20 system.

$\mu$  -Rhythm is blocked by movement, which may be active, passive or reflexive. The blocking effect is bilateral but more stronger in contralateral side i.e. if left hand is moved, the blocking is stronger in the right side. This blocking appears before actual movement of the muscles, therefore it seems to related to conceptual planning of the movement. The  $\mu$  -rhythm is also blocked by light tactile i.e. touching the skin lightly, so some researchers consider the  $\mu$  -rhythm to be the 'idling' state of the sensory cortex and its vicinity.

## Beta Rhythm

$\beta$  -rhythm band consists of basically all frequencies above 13 Hz but practically it is limited to 50 Hz by the measurement limits and to 30 Hz by functional findings. Spatially  $\beta$  -rhythm can be found frontal and central regions. The central  $\beta$  -rhythm is related to *Rolandic  $\mu$  -rhythm* (see (1.2.2)) and can be blocked by motor activity and tactile simulation (planning to move).  $\beta$  -rhythm rarely exceed amplitudes of  $30 \mu V$  and as a rule of the thumb when frequency increases the amplitude decreases and vice versa.  $\beta$  -rhythm is usually associated with increased arousal and activity.

## Theta Rhythm

A band originally a part of the delta band is of frequencies from 4 to 7 Hz.  $\theta$  -rhythm has gotten its name from the presumed origin, the thalamus. EEG of a normal adult consists little  $\theta$  -frequencies and no organized  $\theta$  -rhythm. However, the  $\theta$ - frequencies and  $\theta$  -rhythm play important part during the childhood and in states of drowsiness and sleep.  $\theta$  -rhythm is associated with marking *the maturity of the mechanism linking the cortex, the thalamus and the hypothalamus*. Also it is linked with feelings of disappointment and frustration. For some people the  $\theta$  -rhythm is present when performing mental tasks e.g. problem solving or visualization.

## Delta Rhythm

This rhythm is detected when the subject is in deep sleep at later sleep periods.  $\delta$  -rhythm has a relatively high amplitude and low frequency, 3 Hz or less.  $\delta$  -rhythm decreases with age and can be a sign of brain abnormality if detected in the awake state.

### 1.2.3 Evoked Responses from the User

In order to communicate via brain activity, the user must be able to control the EEG signal. These types brain activities can be divided to two groups: Evoked responses, which are evoked responses by a sensory stimulus, such as flashing light, and spontaneous EEG signals which occur without stimulus, such as  $\alpha$  or  $\mu$  -rhythm, which the user can learn to control with the *biofeedback* (see section 1.3.3).

The evoked responses can be further divided to three main classes:

- **Evoked potentials (EP)** require specific sensory stimulus. An example of EP is *visual evoked potential (VEP)*. If stimulus is given in a form of a flashing light, the EEG over the visual cortex will have the same frequency as the flashing light. A user can be trained to control the amplitude of the EEG over the visual cortex by *biofeedback*. VEP is very easy to detect, making the pattern recognition easy. But the training time required to improve the control is long
- **Event-Related potentials (ERP)** are DC changes to a discrete event. The ERP is a response to a stimulus or an event and it either coincides or follows it after a short delay. But the ERP can also be detected in the absence of the stimulus if the actual stimulus is anticipated to happen or they may precede voluntary motor responses, i.e. moving a hand without being directed to do so. ERPs are believed to be generated by the brain through extracellular potentials associated with the activity of groups of neurons firing in synchrony.

Examples of ERPs are *P3 or P300* ERP, which occurs 300 ms after a specific stimulus that to subject is told response. The stimulus has to be of *bernouilli* type i.e. more rarely occurring alternative from the two alternatives.

Another example is *slow cortical potentials (SCP)*, which are a large increase of surface-negative cortical DC potential, caused by cognitive processing in the brain lasting more than a second or two. When compared to the shorter latency ERP such as P300, the SCPs reflect more global task-related processes. In figure (1.5) is shown SCP development for linguistic and mental visualisation tasks. Notice how linguistic task activates the frontal lobes (electrodes *F3, F7*) and the mental visualisation the posterior parietal lobes (electrodes *P3, P4*)

- **Event-Related Synchronization (ERS)** and **Event-Related DeSynchronization (ERD)** are the AC changes to a discrete event. More accurately ERD/ERS is blocking of the  $\alpha$  -rhythms due to sensory processing or blocking of the  $\mu$  -rhythm due to motor behavior. ERD/ERS occurs during the cortical information processing due to the increased cellular excitability in thalamocortical systems. The difference between ERD and ERS is that a power decrease in the  $\alpha$  -band responds to ERD and increase to ERS.

ERD/ERS starts few seconds before the actual movement and it lasts few seconds and in order to restore the power levels to the reference level, the period between two stimulus should randomized and no less than two seconds.

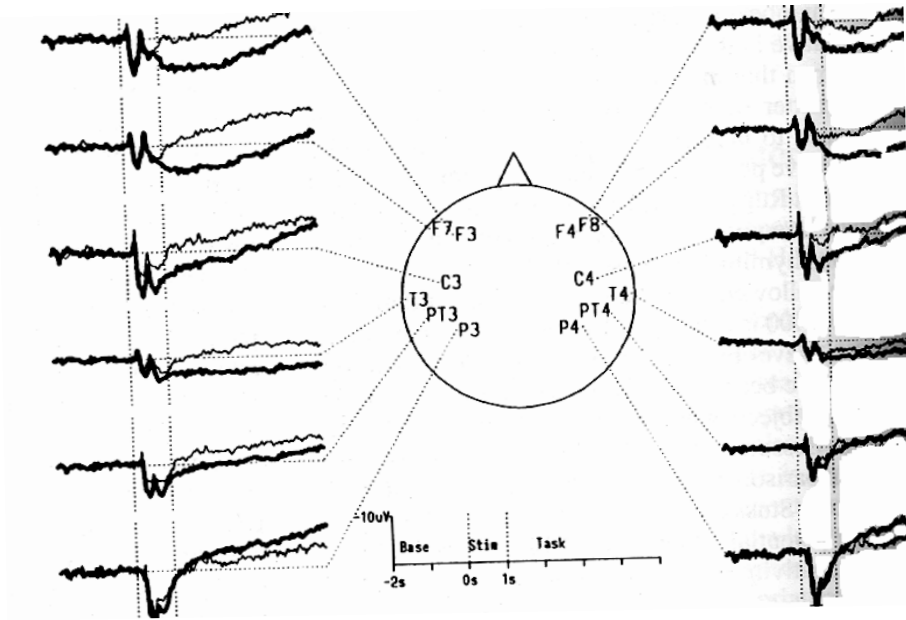


Figure 1.5: SCP development for linguistic and mental visualisation tasks. [Nie99c]

The depth of ERD is affected by the complexity of the task or depth of the attention to the task.

#### 1.2.4 Measuring the EEG signal

The EEG stands for *Electroencephalography* and is measuring the difference in electrical potential between various places on the surface of the scalp. In an EEG measurement a potential difference between two electrodes is measured. Another one of the electrodes may be *passive* in a sense that it is not used to measure the brain activity, but the background electric field of the skin. These electrodes are called *references* and they are attached to ear lobes or mastoids. The placement of the references is delicate, since a reference too close to the brain is corrupted by brain activity and a reference on other parts of the body may be corrupted by muscle's (especially heart's) electrical activity. The signals picked up by electrodes may be combined to *channels* or a channel corresponds to a single electrode. The signal is then amplified and filtered from artifacts and displayed on computer screen.

## Electrode Configurations

The first EEG measurements were done by Berger in 1929 [Ber29]. In 1934 was found that EEG activity varied in different locations of the head, which lead to several different electrode placement configurations in order to achieve the best signal for current experiment. These different electrode configurations of course made the comparison of the results difficult, and therefore in 1958 a common standard, the *10-20* electrode configuration was agreed to be the basic configuration. [Nie99d]. The *10-20* system is shown in figure (1.2.4, left). In this system specific body landmarks are used to define electrode

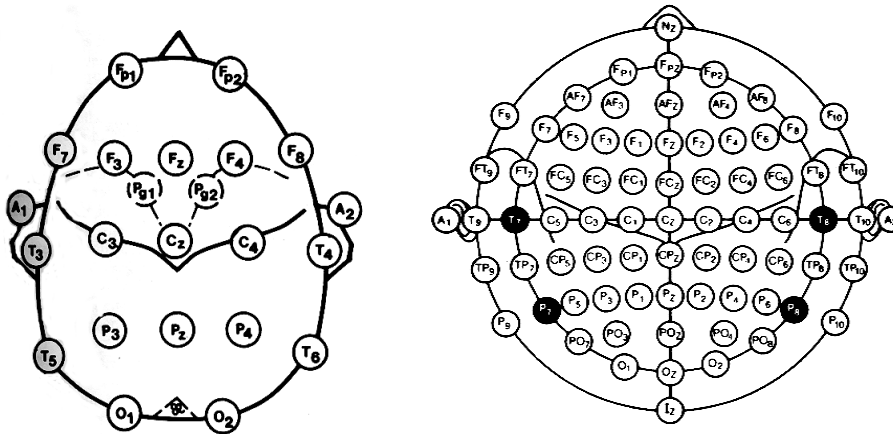


Figure 1.6: On the left: the standard *10-20* electrode configuration. On the right: the *10-20* system augmented to 75 electrode configuration. [Nie99d]

placement, instead of using constant distance between the electrodes. This body landmark based system is easy to replicate in different laboratories and when measuring the EEG of a child, the distances between electrodes grow as the child grows, but the placement remains consistent.

In *10-20* system the electrodes are coded by letter that indicates the anatomic area, and the numbering, in which the odd digits are for the left hemisphere and the even for the right. An exception are the midline electrodes, where the digit is changed to letter z. As can be seen from figure (1.2.4, left), space have been left between the electrodes for additional electrodes if needed, e.g. electrode *F5* can be placed between electrodes *F3* and *F7*. An example of the modified *10-20* system is shown in figure (1.2.4, right), which is a system of 75 electrodes. The number of needed electrodes depend of the type and location of brain activity and the number of channels available.

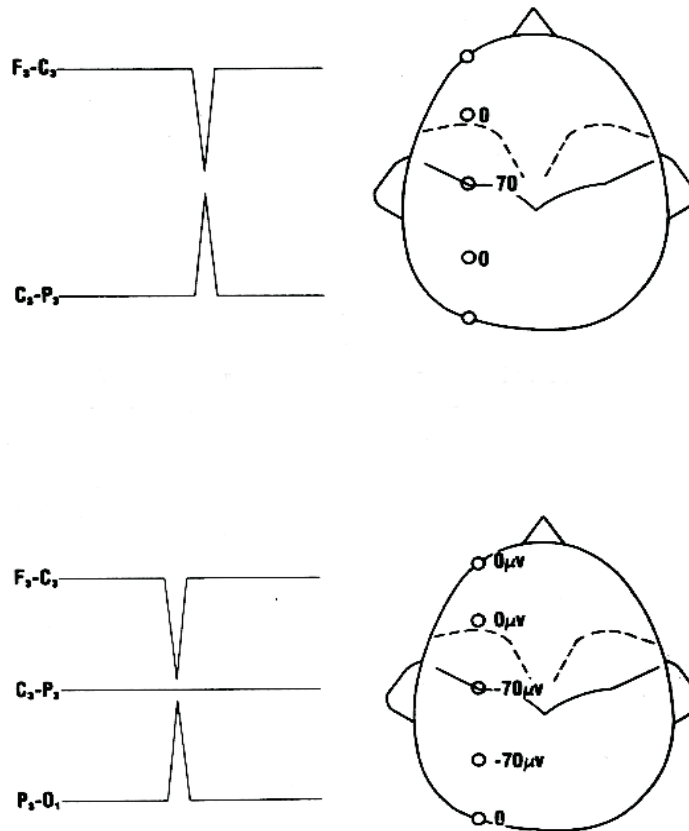


Figure 1.7: Above: Bipolar linkage over two electrodes Below: Bipolar linkage over three electrodes. [Nie99d]

### The Montages

A combination of electrodes used to study particular point in time is called *montage*. The montages have two simultaneous functions: The first is to record the EEG from all areas of the scalp and the second is to record the activity so that it is easily perceived by the reader.

There are two major categories of montages. In the first style, called *monopolar* or *common reference method* approach, all the electrodes are referred to single reference electrode, one common electrode at each side of the head or to combined activity from the two references. The second one is to use scalp-to-scalp linkages where each channel is connected to two or more electrode pairs as shown in figure (1.7). Bipolar montages should only be used if certain type of brain activity pattern familiar to user is sought.

## **Impedance**

Earlier the electrode attachment was tested with measuring resistance rather than impedance, which naturally lead to two problems: The electrodes may become polarized or charged, which may have caused abnormality to the EEG, or if the measurement device used for measuring resistance of electrodes attached to patients head was not specially designed for that purpose, the current was felt by the patient from a tingling sensation to a clear pain.

By preparing the patient well by using high quality paste or gel to enhance the conductivity, impedances of  $3M\Omega$  are usual and improvement to  $1M\Omega$  can be reached easily. An impedance over  $5M\Omega$  should not be approved in any circumstances.

It is important to check the impedances before every EEG measurement because it is not adequate to assume that every time just the routine preparation will produce sufficiently low impedances.

Also it is most likely that the impedances deteriorate during the EEG measurement so impedance check at regular time intervals are recommended. Nowadays many EEG systems have automatic impedance control.

## **Filters**

High -and lowpass filters are used to remove the unwanted part of the EEG signal. The use of the EEG signal varies for different purposes, e.g. for clinical use frequencies 5 - 30 Hz are usually sufficient and research team may be interested in very specific frequency bands, such as  $\alpha$  or  $\theta$  -rhythm.

The biggest difficulty in filter use is that filter affects all frequencies, not just those which are meant to be filtered out. So the art is to set the filter specification to such that unnecessary components are removed but the desired components are not attenuated too severely. Various EEG institutions have recommended that a frequency displayed at 70% or more of its original voltage is acceptable. If the loss is more than a 30% then the distortion is significant and the activity is put outside its frequency band.

## Sensitivity

Sensitivity of an EEG system is defined as a ratio of input *voltage* to output *deflection*. The voltage is the potential difference between two points, and the deflection is vertical distance between two points, that are drawn to computer screen or back in the old days to paper. The unit of the sensitivity is  $\frac{\mu V}{mm}$  and the minimum deflection is set to 5 mm, because this was set when all the EEG system still used paper as an output and 5 mm was the limit which could be seen clearly with naked eye.

Minimum sensitivity is required from an EEG system is usually the minimum electrocerebral activity, i.e. the lowest electrical activity of a human brain. If persons electrocerebral activity is below the limit, which is no activity over  $2\mu V$ , the person is pronounced brain dead. So in order to be able to pick up  $2\mu V$  voltages the sensitivity must be  $4\frac{\mu V}{mm}$ . Today modern EEG system have sensitivity between  $0.5\frac{\mu V}{mm}$  and  $1\frac{\mu V}{mm}$ . As a point of comparison the internal noise of these systems is  $2\mu V$ .

## Artifacts

Artifacts are non-brain based electric fields induces to the EEG, that corrupt and distort the signal, making it unusable or difficult to interpret. There are several categories of artifacts. Maybe the biggest and most commonly encountered are machine and impedance problems. In these cases there are usually broken or improperly attached electrodes. Second major type in this category is the 60 or 50 Hz artifact, which is induced from nearby electrical equipment or *ground loop*. Ground loop occurs when the patient is connected to the ground more than a once and there is a difference between the grounds. Another 60 Hz problem occurs when the reference electrode is directly short circuited to active electrodes, which is particularly nasty problem, since the impedance measurements give correct results and problem stays from one montage to another.

The next biggest source of the artifacts are of physiological origin. Obvious source of oscillation is of course the heart. Heart induces artifacts of three type: The high and steep spike in *electrocardiograph (ECG)* during the heartbeat as the contraction of the heart muscle is controlled by the body with electric signals. The second is the movement of the blood in the veins close to the electrodes which causes electrodes to move. The third is less common and less severe but it has a cool name, *ballistocardiographic artifact*. It is a slight body movement caused by heartbeat. In general all body movements cause artifacts as electrodes move. As electrodes move the impedance can vary rapidly which causes

rapid changes in the EEG voltage. Therefore the patient is always instructed to remain as stationary as possible.

Another physiological artifacts are the eye artifacts. They develop the biggest distortions but are easiest to document in clinical use. There are several method for removing the eye artifacts automatically, e.g. [HO99] or [Bir00].

When considering all these sources of artifacts one can realize how difficult problem an on-line BCI system can be. If these BCI system are meant to help people to live normal everyday life the EEG will be heavily corrupted by the artifacts as people move around and they get close to various sources of electric fields, e.g. mobile phones. Therefore effective measures have to be taken to shield the system from unwanted sources and in general, the system have to be robust and to be able to deal with even poorer signal to noise ratio, which is in the EEG already small.

## 1.3 Brain Computer Interface (BCI)

In this section the basic principles of a BCI system are discussed and a BCI system called *Adaptive Brain Interface (ABI)* is described.

### 1.3.1 What Is BCI

What makes the BCIs so distinctively different from other *human computer interfaces (HCI)* is that in a BCI, no peripheral nerves or muscles are needed for communication, only brain activity is required. This system is an user interface. This means that it can be applied to interact with surrounding world by controlling computer or wheelchair or by verbal communication by producing letter from a virtual keyboard on a computer screen.

Naturally those who most benefit of this kind of a system are those with severe disabilities such as *amyotropic lateral scleroses (ALS)*, *severe cerebral palsy*, *head trauma*, *spinal injuries* or *anyone with his intellectual capabilities remaining but physical capability severely reduced*.

However, everyone can benefit from these kinds of systems. The first applications will probably be control and communication aids while working in difficult and hostile environments or perhaps as a entertainment purposes from videogame controller to a ultimate TV remote control (no need to move even that one finger). But it can be expected that the applications of the BCI will be in daily use in the future with a variety of applications.

Development of the BCI system can be considered to have started from the very beginning of the discovery of the EEG was, in 1929 [WV00]. The first actual applications were developed in 1970's and today there are at least 22 active research groups all over the world [WV00].

### 1.3.2 System Components

A BCI system can be divided to four components. The first is the settings of the detection equipment, which is basically always based on the EEG techniques. The second is a system or an algorithm to provide a stimulus. The third is detecting the response for corresponding stimulus and the fourth is the actual control of the application of this interface.

## Set-Up of the System

Depending of the BCI system, the number of used electrodes to measure the EEG varies from one to the full *10-20* system. Usually certain amount of calibration or baselining is required to tune the system to a predefined status. This tuning may include redefinition of electrode location, EEG measurement parameters e.g. impedance of the electrodes and some general thresholds and weights of the system in question.

## Operation

Operation of the BCI system is not simply listening the EEG of user in a way that let's tap this EEG in and listen what happens. The user usually generates some sort of mental activity pattern, that is later detected and classified. These activity patterns usually stimulate the upper cortex of the brain where a good quality EEG is attainable since the attenuation as it travels through the skull is less severe. The mental activity patterns or task varies from stimulating visual [Sut92] to motor cortex [Pfu93] for controlling  $\alpha$  -and  $\beta$ -rhythms.

## Detection

The detection of the input from the user and them translating it into an action could be considered as key part of any BCI system. This detection means to try to find out these mental tasks from the EEG signal. It can be done in time-domain, e.g. by comparing amplitudes of the EEG [Bir00] and in frequency-domain, e.g. by studying power spectra [Pfu00], [PRS99]. This involves usually digital signal processing for sampling and bandpass filtering the signal, then calculating these time -or frequency domain features and then classifying them. These classification algorithms include simple comparison of amplitudes [Wol91], linear and non-linear equations [Pfu93] and [RP99], artificial neural networks [Hir90] etc. This is the part where the system *adapts* to the user. By constant feedback from user to the system and vice versa, both partners gradually learn more from each other and improve the overall performance.

What is crucial in the classification of the mental tasks is to avoid *false positives*, i.e. wrong classification, which can have terrible effects. Consider for example that user controlling a wheelchair by a BCI system is sending an order to the wheelchair to stop but the chair just keeps going forward. Therefore the best algorithm to classify the tasks should strike a balance between a high classification rate, low proportion of false positives and

time required for single classification.

## **Control**

The final part consists of applying the will of the user to the used application. The user chooses an action by controlling his brain activity, which is then detected and classified to corresponding action. Feedback is provided to user by audio-visual means e.g. when typing with virtual keyboard, letter appears to the messagebox etc.

### **1.3.3 Training**

What should be emphasized is that this operation of a BCI system should be considered as a new skill, which must be practiced basically as any new skill from writing to playing an instrument. This training begins with very simple exercises where the user is familiarized with mental activity which is used to relay the information to the computer. Then when moving on adding further depth to the training by introducing more patterns, complexity of application and requiring improved performance. The training is the part where the user *adapts* to the BCI system. This training is subject to all the same issues that have effect wheter you are training to eat with sticks to drive a car. Motivation, frustration, fatigue, etc. apply also here and their effect should be taken into consideration when planning the training procedures. The opinions from the users naturally help to alleviate this problem but the value of these are not clear, as BCI training is a very personal experience

What makes this communication with brain activity so special is that the training depends solely on user. For example, when you train to eat chinese food with sticks, someone can put your fingers to the right position, i.e. you get *feedback* from a teacher. But nobody can put your brain activity as it should be, it has to be done by yourself. Therefore providing feedback to the trainee is essential.

## **Biofeedback**

The feedback in BCI systems is *biofeedback*, also called *operant conditioning*. The definition of the biofeedback is *biological information which is returned to the source that created it, so that source can understand it and have control over it* [O'H98]. This biofeedback in BCI systems is usually provided by visually, e.g. the user sees cursor moving up

or down [Wol91] or letter being selected from the alphabet [LE88]. Biofeedback is used essentially in training but it is also present at the actual use of an application, although benefits of this are not quite clear [McF98].

### 1.3.4 Adaptive Brain Interface

The BCI system analysed in this work is a system called *Adaptive Brain Interface (ABI)*. ABI project is funded by EU and developed in co-operation by

- Institute for Systems, Informatics and Safety, Joint Research Centre of the European Commission.
- IRCCS Rehabilitation Hospital S. Lucia, Italy
- Fase Sistemi Srl, Italy
- Laboratory of Computational Engineering, Helsinki University of Technology, Finland

Figure (B.1) shows a component diagram of the current ABI system.

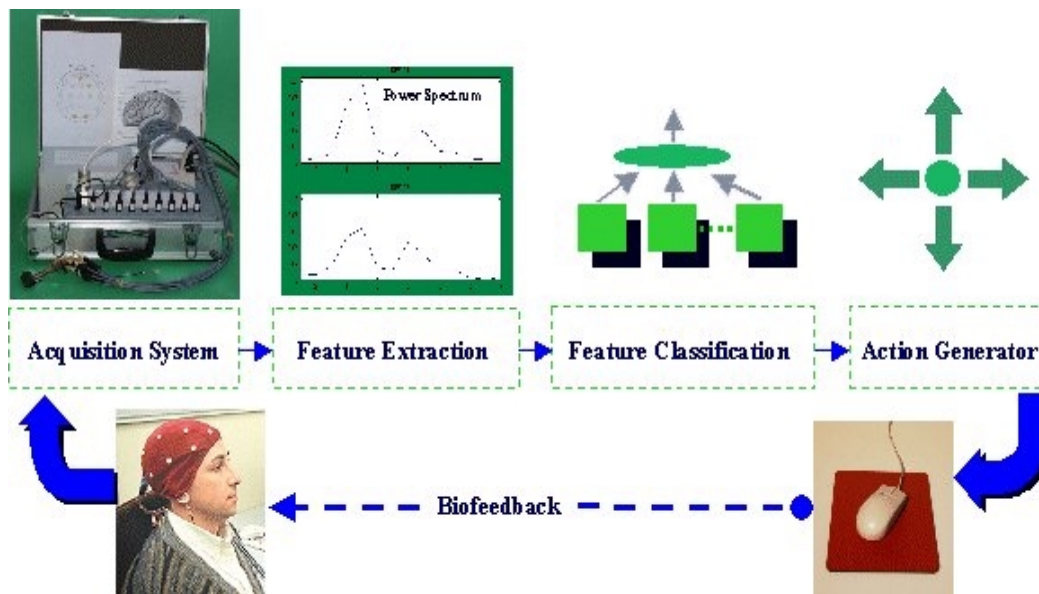


Figure 1.8: The key components of the ABI

ABI system consists of portable EEG acquisition system, digital signal processing card and a regular computer. In the near future, the current EEG system will be replaced by

totally *wearable* one.

The approach on which the ABI is based, as the name implies, is the *adaptiveness*. That means that both the system and the user adapt to each other as explained before. In ABI the adaptive part is the *local neural classifier* [dRM00], which is responsible for classifying input signal, and the user adapts by training the chosen the mental tasks which he/she finds most comfortable and effective to use.

Second important approach is that this system should also work reliably outside laboratory environment, i.e. in normal everyday life. This calls for an easy to use, wearable (small and light), comfortable and robust EEG equipment.

## Overview of the System

In ABI system the EEG signal is measured from the electrodes *F3, F4, C3, C4, P3, P4, T5, and T6* of the standard *10-20* system (see figure 1.2.4, left). The EEG is digitized and sampled at a rate of 128 Hz. A segment of 0.5s long is called a *sample*, and is taken from all the channels. Then these samples are filtered spatially by applying *surface Laplacian* derivation (see 2.2.1) over the six centro-parietal electrodes over their neighbours. Finally the samples are bandpass filtered with 2nd order Butterworth filter with passband of 4-45 Hz. The features used are power spectras in decibel scale estimated by using *Welch periodogram* algorithm, using *Hann* windowed 1s segments (i.e. two samples) with 0.5s overlapp.

## Training

In studies concerning training procedures for ABI, we have found that there are no universal features that could be applied for every user since our individual brain functionality and the level of activity is unique. Therefore it is important that both the user and the system adapt to each other. Another issue is the brain activity used for control. Many other BCIs (see section 1.4) concentrate only to one type of brain activation, such as ERD/ERSs (see 1.2.3). We have chosen different approach, where no single activation type is followed but the overall activity level in various parts of the brain. This approach, as our coordinator puts it, is *is closer to real life as subjects are not passive but make decisions spontaneously and self-paced*. The most commonly used mental tasks we use to stimulate different parts of the brain are as follows:

- Relaxation: Total physical and mental relaxation (closing eyes, relaxing muscles etc.). Alpha -rhythm appears at the parietal, posterior and occipital lobes.
- Imagined movement of left or right hand: This stimulates contra laterally (imagined left stimulates right hemisphere) the motor center in the parietal cortex, introducing ERDs and blocking the alpha -rhythm.
- Mental arithmetics: User starts to make subtractions like  $63-2 = 61$ ,  $61-2 = 59$ ,  $59-2 = 57$  etc. in his/hers mind. This stimulates the frontal lobe.
- Visualizing a 3-D cube and rotating it in one's mind: Activates the visual center at the occipital lobe.
- Resting (eyes opened and not actively thinking anything): Weak Alpha -rhythm may appear to the parietal and posterior lobes.

Of course it may not seem to be natural to direct mouse to the left by subtracting numbers but this approach gives total *autonomy* to the user. The user is not dependent of any outside help such as instructions or flashing lights.

When compared with other BCIs, one of the ABI's areas of good performance is the time required for training. User can acquire good control over the system just in five days, one hour per day. But the basic skills can be learned even faster, as was shown during *IST2000* fair in Nice, October 2000, where three members from the audience asked to try the system and just in one hour they trained themselves to type few letters and to play a little bit of the pacman game.

## **Performance**

Currently ABI is used to tell apart three different tasks i.e. user has three command at his disposal. In the near future this amount of commands is upgraded to five e.g. control of a computer mouse (up-down, left-right and click).

The recognition rate for the three tasks is around 70%, which may seem to be modest. However, because the false positive rate is only 5% and the classification is made in every 0.5s, a reasonable robustness is acquired and the response time is feasible.

### 1.3.5 Applications

In figures (1.9), (1.10) and (1.11) are shown some of the current applications of the ABI system. The first one is a *virtual keyboard* designed as an example of a communication channel for disabled people. The keyboard quite simple, a letter is selected by dividing the keyboard to three segments with each selection until only a selected letter remains.



Figure 1.9: Virtual keyboard application for producing text

Figure (1.10) shows robot system, which is used to model control of a wheelchair. Here the user gives commands to robot via ABI and robot the moves to desired place. Important future aspect for these wheelchair applications is to add lower-level intelligence to the wheelchair. This means that e.g. only the general direction is needed by the wheelchair and it avoids obstacles by itself on its way to given direction.

On the last figure (1.11) an entertainment application is shown, where by mastering two mental tasks, the user can play classic *pacman* game. The two tasks used so that the game gives the player two possible directions where to turn according to the pacmans position in the maze.



(a) The robot which is used to model the movement of a wheelchair



(b) Robot being controlled with ABI

Figure 1.10: Robot application for ABI

## 1.4 Other BCI systems

Here some BCI systems are introduced. They are divided by the type of brain activities described in 1.2.3 used for control. This is by no means an exhaustive collection of papers and more information can be e.g. found in [WV00] and [IEE00].

### 1.4.1 VEP Detection

BCIs based on *visually evoked potentials (VEP)* are not truly BCI because user needs muscle control to turn his eyes to a certain point. But for comparisons they are included here.

Three groups have studied the VEPs applicability for BCI. Sutter's group [Sut92] used a matrix of 64 kernels on a computer screen. The kernels were normal alphabets and commonly used english words. When a subject selects by looking a letter from the screen the english words on the matrix are changed to word beginning with that selected letter. Sutter switched 64 screen positions between red and green with lengthy binary sequence or in some trials by reversing a checkboard pattern. Each screen position was shifted



Figure 1.11: Classical Pacman game being played with ABI

20ms in the binary control sequence relative to its neighbour, and the entire sequence was autocorrelated with the VEP, which response lasts about 80ms, in overlapping increments beginning 20ms apart. The resultant vector was stored in a 64-position array of registers. When a coefficient remained greater than certain threshold and all the other coefficients for a certain time period, it was considered to have been selected by the subject. The VEP was measured with electrodes placed over the visual cortex at back of the skull. Experiment suggested that electrode placement and stimulation mode should optimized individually for each user for good target discrimination. This system was evaluated by seventy healthy subjects, who achieved response times from one to three seconds after training period of 10-60 minutes.

Cillier's group [Ci193] used as a stimulus a series of four lights, whose intensity was varied with a 10Hz sine wave in phase quadrature. The subjects' VEP was detected and measured by two electrodes located at occipital lobe. The four lights were located at corners of a computer screen and each light corresponded to a row of a virtual keyboard. By choosing a certain light, the corresponding row was selected and further split to four parts. The subject could reach any letter on average of three selections, and it took about 15 seconds. Initial training period consisted concentrating to each light for a five seconds.

In 1995, McMillan and Calhoun used VEP to control the roll of a plane in a flight simulator. Again the VEP was measured with two electrodes over the visual cortex and it

was generated with 13.25Hz sinusoidal light. The magnitude of the 13.25 Hz component of the EEG was measured with a lock-in amplifier and when ever the magnitude (amplitude) exceeded certain threshold, the simulated plane rolled right, and if the amplitude was lower than another threshold, the plane rolled left. After 12 hours of training, subjects were able to respond to random turns with accuracy of 80%.

In a separate experiment, the knee flexion was controlled by VEP. By using the same experiment set-up as above and a stimulator attached to anterior surface of the thigh over the rectus femoris muscle. Three subject were trained to use this system and after 5 hour training 96% of attempts to extend the knee from perpendicular position to a horizontal position while sitting in a chair were succesful. Time required to extend the knee was five seconds and to lower it back took six seconds.

## 1.4.2 P300 Detection

This system detects subjects P300 component from subjects event-related brain potential [LE88]. The P300 is a positive-going ERP with a latency of 300ms and it is triggered by a rarely-occurring stimulus that the subject is instructed to detect. This stimulus has to be of *Bernoulli* event i.e. one of two types, e.g. from two lights one flashes regularly and the other rarely. The observation of the rarely flashing light starts the P300.

The EEG measurement was done by using a single electrode placed at  $P_z$  of the standard 10-20 system. The received signal was bandpass filtered to 0.02 - 35Hz and sampled at 50Hz.

The P300 was evoked by using “odd-ball” paradigm, where subject is asked to pay attention to some rarely happening stimulus by some non-motor way, such as counting the occurrences. This stimulus could be e.g. a light normally blinking green to suddenly blink red. A reliable detection of P300 requires averaging several samples of EEG. The purpose of this work was to determine the required number of presentations of the stimulus in order to reliably detect the P300 from the EEG. The used system was a 36 part virtual keyboard on a computer screen consisting of alphabets, digits and some other normal characters of a keyboard. The rows and columns of this keyboard were made to flash randomly, starting with rows, at intervals of either 125ms or 500ms. The subjects were asked to concentrate to one letter of the word *brain* and count the times when a row or a column of that letter flashed on the screen. Results showed, that 30 stimuli were needed at 500ms intervals and the time needed to produce one letter was 93.6 seconds. When the stimulus

was applied at intervals of 125ms, only one stimulus was needed and the time needed to present a letter decreased to 1.245 seconds.

P300 detection from the EEG signal was studied with four different techniques: stepwise discriminant analysis, peak picking and covariance. The stepwise discriminant analysis gave the best results of 95% accuracy to select a character from a set of 36 characters within 26 seconds.

### **1.4.3 Slow Cortical Potentials (SCP)**

In [Bir00] is described a “*Thought Translation Device (TTD)*”, which is a BCI based on SCPs (see section 1.2.3). In this approach the user learns to control the amplitude of his/hers SCP. This device is based on experiments with five subjects at their homes. The EEG recording were made with portable EEG device using total of eight electrodes from frontal, central and parietal sections of the head. EEG signal was sampled at frequency of 256 Hz and filtered from eye-artifacts. Visual feedback is presented to the subject in a form of a “ball shaped light” that moves away or towards a target, that is located at center of the computer screen, depending the level of the control at a given time. Also, the ball is highlighted when the subject has to produce negative SCP and flashes when positive SCP in needed. For the classification, the EEG is averaged in windows of 500ms moving in steps of 63ms. When the change in the amplitude is detected, it is visualised to the subject by a smiling face and a new trial begins. A training day usually consists 6-12 five to ten minute long training sessions and consists of 70 - 100 trials. Subjects are trained several times per week. The training procedure is a kind of shaping program, where the required amplitude change i.e. the amount that the user has to increase/decrease the amplitude is gradually increased from  $5\mu V$  to  $8\mu V$ . When the subject has stable performance of at least 75% of the trials correct, then he moves to the first level of the language support program. In this level I language support program, a letter is selected from an alphabet displayed on the computer screen by dividing it to two part at every selection. So the alphabet gradually has only two letters to choose from and after the final selection the letter is added to current word that the subject is producing with the program. After the first two letters being selected, a erasing option is added to the screen for the user to edit out letters from the text field.

From the five subject who used this system the first was trained total of 260 sessions and after 100 sessions (sessions consisting 70 - 100 trials) the subject has used the program with accuracy between 70% and 80%. The second subject has been trained for 220 ses-

sions and his classification rate is about 80%. For the both the trial length is about 4.5 seconds and with median of 28 trials to select a letter, it takes about two minutes to select a letter with the program.

#### 1.4.4 EEG $\mu$ -rhythm Conditioning

Here the user controls amplitude of a  $\mu$ -rhythm component in the EEG.  $\mu$ -Rhythm is detectable pattern in the EEG at 8-12Hz frequency range, centered about 9.1 Hz. The  $\mu$ -rhythm is *Event-Related Desynchronization* (ERD) that is desynchronized by movement, tactile stimulation or planned movement. This approach is based on Kuhlman's [Kuh8Aa], [Kuh8Ab] approach and three groups have studied  $\mu$ -rhythms applicability to BCI.

Wolpal's group [Wol91], [Wol00] detected subjects  $\mu$ -rhythm amplitude from the square-root of spectral EEG power at 9hz, using two electrodes located at near C3 in the standard 10-20 system. Experiment set-up was such, that the subject tried to move cursor to a target placed randomly at somewhere at the top or the bottom of the computer screen. Cursor step size was varied by the operator of the system. Commonest i.e. easily achieved  $\mu$ -rhythms ( $< 4\mu V$ ) left the cursor in place or moved it downwards, while higher  $\mu$ -rhythms ( $> 4\mu V$ ) moved the cursor upwards. Observations made during the experiment were that there was no relationship between subjects muscular (EMG) activity and the  $\mu$ -rhythm. Also no connection was made between the subjects visual  $\alpha$ -rhythm and  $\mu$ -rhythm. Experiment lasted two months, consisting 12 45-minute sessions and the results were such, that from the five subjects participating four acquired impressive control over their  $\mu$ -rhythm. Classification rate was 80-95% and time required varied between 10 to 29 hits per minute.

McFarland [McF93] used similar setup but with tighter time constraint. In this experiment the target moved from left to right side of the screen in eight second and subject had to move the cursor to one of the five vertical positions to intercept the target. Results were that all four subject were able to reduce the distance between the cursor and the target and three out of four had significant success.

In Wolpal's group's later experiment in 1994 [Wol94] both left and right ERD were measured by two electrodes that were placed upon the left and the right side of the sensorimotor cortex. Power spectrum of width of 5 Hz and centered at 10 Hz was calculated from the signal at every 200ms. These powers of the  $\mu$ -rhythms were then converted to both horizontal and vertical cursor movements by linear equations relating vertical movement to the sum of the signal and horizontal to the difference of the signal. The task was to move

the cursor from the center of the screen to randomly chosen corner. Subjects were trained for a 6 - 8 weeks and the best subject hit 70% of the targets in average of 26 minutes.

Pfurtcheller's group [Pfu93] used contralateral blocking of the  $\mu$ -rhythm during the 1-second period prior to motor activity, which in this case was pressing a microswitch using either right or left index-finger, to predict which response was to follow. 30 electrodes were evenly placed across the sensorimotor cortex, two of them located at *C3* and *C4* in the standard *10-20* system. Used feature vector for classification was power spectrum acquired by *Hilbert transform*. In the initial training all the 30 electrodes were used but in the subsequent trials only the electrodes located at *C3* and *C4* were used. Power spectrum was calculated from five time points and used classifier was *Learning Vector Quantizer* type neural network. Classification accuracies of 90% were reported.

In 1994 [Nie99b] studies with on-line *Graz BCI I* system showed that ERD can be modified by biofeedback. In these sessions a subject after the first session had a classification rate a of 70% which rose to 85% after three more sessions.

Also in 1994 [Nie99b] Pfurtcheller's group showed that different ERD is produced when moving different parts of the body. Because the hand and finger movement is accompanied by blocking of the  $\mu$  -rhythm (10 Hz ERD) and by short lasting  $\gamma$  -rhythm (30-40 Hz ERS), they can be told apart from other body parts. Power spectrum estimates were calculated from bands 10-12 Hz, 30-33 Hz and 38-40 Hz in every 250ms from the EEG measured with eight electrodes in a rectangular array over the sensorimotor cortex. Introduction of the  $\gamma$  -rhythm improved the classification rates from 58% to 70%.

This knowledge is adopted to *Graz BCI II* which uses movements of right index finger and foot and left index finger to generate ERDs. The EEG has three bipolar channels received from six electrodes over the sensorimotor cortex. Again the power spectrum estimates are calculated in every 250ms and from one second period samples from all three channels are concatenated and used as an input to a DSLVQ classifier. The system was tested by four subjects in four one and a half hour sessions over a two week period. Each session consisted of four blocks of 60 trials with five minute break between each block. Set-up was such that one second after a cue, in this case a beep, an arrow appeared on a computer screen pointing left, right or down. After 1.25 seconds, the subject pressed switch with corresponding finger or raised the right foot toes. The EEG signal one second before the movement was classified by the system and classification rates varied between 77% and 81%.

More recent research by Pfurtcheller's group [Pfu00] has been done in feature extraction

and classification. The following feature extraction methods were studied:

- *Power spectras from subject-specific frequency bands*, where from the initial training data (the training without feedback) the most reactive frequency bands are determined by using *distinction sensitive learning vector quantizer (DSLQ)*. This method adjusts, by weighted distance function trained with supervised learning, the influence of the different frequency components. The outputs are the weight values of individual frequency components according to their relevance for classification.
- *Adaptive autoregressive (AAR)* features, estimated with recursive least squares (RLS) algorithm. The main benefit of this method is that no assumptions about the frequency bands has to be made. AAR parameters were calculated separately from the EEG channels *C3* and *C4* and they were linearly combined.
- *Calculation of common spatial filter (CSP)*, which reflects the specific activation over the cortical areas during hand movement imagination. From the EEG of the weighted electrodes spatial patterns are extracted for both the left and right hand movement imagination. This is based on simultaneous diagonalization of the two covariance matrices. The patterns that maximize the difference between the left and the right are selected by choosing the patterns where the variance between the two states varies most during the operation. During the on-line operation the incoming EEG is filtered with the most important filters and the variance is calculated from several consecutive samples.

The following techniques for the classifications were considered:

- Neural network (DSLQ) based classification [Pfu96]. In this method the one second long EEG samples were further divided to four 250ms samples, from which the predefined power bands are extracted. Then these four samples are averaged and classified. Correct classification results between 10- 17% were recorded.
- Linear discriminant analysis (LDA), which was used with AAR and CSP features. LDA is suitable classification method when the dependent variable is non-metric and the independent variables are metric. In LDA the sum of weighted independent variables for each sample is calculated. This sum is called discriminant score and when a sample is being classified, its discriminant score is compared to the class prototype scores and classified to the closest score. The weights can be solved in several ways, such as the method of least squares. More detailed explanation can

be found e.g. [Bis95]. Classification rates for three subject varied between 1.8% - 12.5% when using CSP features and between 5% - 9% when using AAR features.

After experimenting with these techniques [Pfu00], Pfurtcheller's group came also to conclusion that both the machine and the man must adapt to each other and the importance of the biofeedback during the training is essential. Especially instantenous feedback provides better result compared with e.g. feedback after five seconds.

Another intresting application from Pfurtcheller's group [Pfu00] is remote control via the Internet, so the patient can stay at home, which is important issue since many of the user are severely handicapped, and the operator/doctor/trainer can remain in hospital or where ever he/she may be working.

### **1.4.5 EEG Pattern Recognition Approach**

Idea behind this approach is to take a kind of 'photograph' what is happening in the brain right now, instead of carefully waiting, timing and locating certain brain activity. Of course the ERDs, EPs etc. are used here also but rather as a combination than a single events that are searched.

Hiraiwa's group [Hir90] studied readiness potentials (RP) of the subjects after they had pronounced one of five japanese syllables or moved a joystick in one of the four directions. EEG signal was collected from 12 channels (*Fp1, Fp2, Fz, C3, C4, Pz, F5, F6, F7, F8, O1 and O2*) of the standard *10-20* system. Data was classified by two *multilayer perceptron (MLP) neural networks*. The first was trained with average data and the second one for single-trial, on-line purposes. For the averaged data, the results were high, but for the on-line application they were less reliable.

Keirn and Aunon [Kei90] set out to find the best signals that could be differentiated by their EEG signal. They used *Bayesian quadratic classifier* to classify one of the five mental task that the subject was concentrating. These tasks were *relaxing, non-trivial multiplication problem, visualizing the rotation of three dimensional object in one's mind, mental composition of a letter to a friend and a visualizing numbers being written to a blackboard in increasing order as the previous is wiped before new is written*. EEG data was recorded from six electrodees placed on *P3, P4, O1, O2, C3, C4* of the standard *10-20* system. The features were power spectra extracted by using *Wiener-Khincine* method and later *auto-regressive (AR)* method, which proved out to be better. Between any mental

tasks and a resting state, the classification results were between 80- 90%.

Penny's and Roberts' approach [PRS99], [RP99] is based on real-time analysis of a single EEG channel, located between and 3 cm posterior to the standard *10-20* system electrodes *C3* and *C4*, which lie over the primary motor cortex. The subjects tried to move a cursor vertically in a computer screen to target icon, which was located either upper or lower part of the screen. Each attempt to reach the target last 10 - 15 seconds. The mental tasks used were arithmetic operations for the upwards movement and imagined movement of the dominant hand for the downwards movement. Training of the system was such, that every attempt to reach target was used as a training set for the next attempt. Initial training sets were recordings of 10 second for both tasks.

Features were extracted by using 8th order autoregressive model. The model order was estimated from a 20 second section of the EEG data using Bayesian scheme. Used algorithm for calculating the AR model was *lattice-filter* approach

The classifier was *Bayesian logistic classifier*. In their previous work [PR98] they had found that a simple linear classifier was only marginally worse than more computationally intensive methods, such as artificial neural networks, so a linear model were chosen for its speed and minimal loss of performance. Weights of the classifier were resolved by using *Bayesian evidence framework* [Bis95].

To improve decision making further they used *moderation* process, which moves posterior probabilities closer to prior probabilities if the variance of likelihood of the weights starts to rise, indicating reduced decision certainty. Also *latent-space smoothing* was used, which takes account also previous decisions and probability threshold of 0.6 was used.

Total of seven subject tested this system and produced 140 results. These results were acquired in five blocks of 6 attempts, one attempt being both move up and down, first attempts being initial training data recording. Three rejection scenarios were used. In the first all three method for controlling the classification mentioned above were used and if over 50% of an experimental block were rejected, then the whole block was rejected. In this scenario 87% of all classifications were correct, but 21% of the blocks and 28% of the remaining data were rejected. The second scenario is same as the first, but without block removal. Classification rate was 76% and rejection rate was 34%. In the third one no rejection methods were used and classification was made sample-by-sample, classification rate being 53%.

Also certain *control blocks* were recorded from the subjects, in which the subjects were

asked to relax and not to think either task.  $\chi^2$  statistical test showed, that 86% of these runs were indistinguishable from the random classifier to the  $p \leq 0.05$

# Chapter 2

## Statistical Modelling

The main purpose of this work was to study methods to classify the EEG signal to corresponding tasks. This is done by a *statistical model*, or more suitable in this case, a *classifier*. The purpose of a classifier is to learn a process that produces the outputs to be classified from observations of the process. Therefore before the classifier can start classifying the output for a given input, it has to be *trained* for this given system. This training is usually done by adjusting classifier's parameters by some learning rule, which is usually non-linear. This makes the training a computationally expensive procedure. The key issue in training is that we do not want the classifier to learn the data that is has been given, but the process that generates the data. This is called controlling the *complexity* of a model/classifier. We do not want the classifier to be too complex, when it pays attention to all the details, even the furthest outliers, and having no *generalization* capacity. On the other hand, if the classifier has very low complexity, it will make a gross generalization about the process, e.g. following average, without being able to capture the actual process. The used classifiers in this work are *Multi-Layer Perceptron Neural Networks (MLP)* and *Radial Basis Function Neural Networks (RBF)*

Before the signal is classified, it is often convenient to transform it to more suitable form, e.g. by dividing it to segments, reducing its dimension, extracting only meaningful information etc. In this work the *power spectrum* of the signal is used as a feature. The power spectrum tells the frequency content of the signal, so from it can be observed what EEG frequency bands are present. This information is combined with spatial information given by the location of the electrodes by combining the spectra of all used electrodes to one feature. The power spectra were calculated using *Welch method*, *Blackman-Tukey method*, *Autoregression model's transfer function* and *wavelet coefficients*.

In order to extract as good features as possible, it is often useful to preprocess the signal. In this work the signal was first transformed to reference free signal via *surface Laplacian*, then it was bandpass filtered by using FIR -type lowpass-highpass filter pair. Finally, white noise was removed by applying *wavelet denoising based on NML -criterion*.

In classification point of a view, the classification of the on-line EEG signal seems to be a very difficult problem. This is illustrated in figure (2). The means of all three classes (see section (2.1) are almost equal and *they are well inside the standard deviations of each other*, except the class 1 at location of the electrodes *Cz, C4, P3* and *P4*.

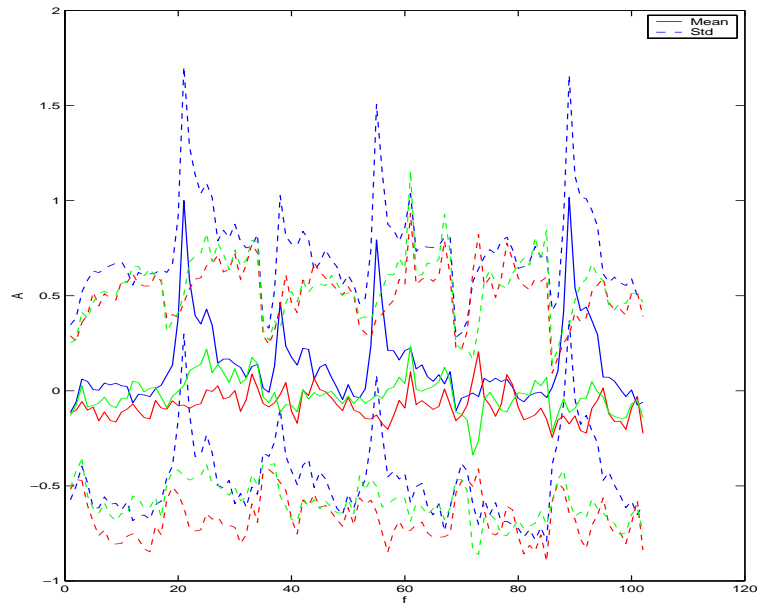


Figure 2.1: The problem of high variances of classes 1 (blue), 2 (red) and 3 (green)

## 2.1 Used Data

The used data is an EEG recording from a single person following the measuring protocol of the ABI system described in section (1.3.4). The recordings are summarized in table (2.1) and the used mental tasks in table (2.1). The EEG signal was samples at sampling frequency of 128 Hz and it was processed in 0.5s segments called samples. *Only three classes were studied in this work and they were classes 1, 2 and 3, corresponding to mental tasks relax, cube rotation and imagined movement of left hand, respectively*

Session	Date	Time	Class 0	Class 1	Class 2	Class 3	Total
1	19 Oct 1999	10:45-11:30	544	270	1088	612	2514
2	20 Oct 1999	16:45-17:30	383	258	429	851	1921
3	21 Oct 1999	17:15-18:00	391	158	588	422	1558
4	26 Oct 1999	15:00-15:30	249	163	195	239	845

Table 2.1: Summary of the recorded EEG data

Class	Mental Task
0	Resting
1	Relax
2	Cube rotation
3	Movement of the left hand

Table 2.2: Used mental tasks

### 2.1.1 Baseline removing

These samples were further preprocessed by *removing the baseline*. The purpose of the baseline removing is a kind of normalization since the overall level of the brain activity varies from day to day [O'H98]. The baseline was created by averaging the first forty 'resting' (see section 1.3.4) samples of each training session and that average was then *divided* from the rest of the samples (subtraction in the logarithmic scale). The variation in brain activity is assumed to be *multiplicative* instead of *additive* [Nie99e], [dRM00], therefore the use of the logarithmic scale.

### 2.1.2 Removing of the samples near class transition

When extracting Fourier spectral features using the Welch method (see section 2.3.1) data segments are averaged. When class transition occurs these segments may correspond to two different classes, making classification impossible. Therefore sufficient amount of samples were removed before and after the class transition to make sure that the classes do not mix up.

## 2.2 Preprocessing

The raw EEG signal requires some preprocessing before the feature extraction. This preprocessing include removing unnecessary frequency bands, averaging the current brain activity level, transforming the measured scalp potentials to cortex potentials and denoising. The main goal of the preprocessing is to transform the signal into a predefined state most suitable for feature extraction, should the outside circumstances e.g. measurement enviroment change.

### 2.2.1 Laplacian filtering

A major improvement in the quality of a recorded EEG -signal can be achieved by transforming a conventional reference-dependent signal into a reference-free signal. Benefits of the reference-free signal are that they are free from the effects of the electrical reference, most importantly spectral dependence of the reference electrode used for measurements. Also reference-free potentials are not blurred by the skull.

*Surface Laplacian* [F.96] is the second spatial derivative of the scalp potentials along an approximation to the scalp surface and it provides an estimation of cortical potentials from the scalp potentials. This work uses planar approximation of the scalp surface and the reference-free potentials are calculated with equations (2.1) for the electrodes  $C3$ ,  $Cz$ ,  $C4$ ,  $Pz$  and with (2.2) for the electrodes  $P3$ ,  $P4$  of the montage used in the ABI system, see 1.3.4 and [dRM00]. Notice that electrodes  $F3$ ,  $F4$  are discarded after the Laplacians are calculated.

$$u_{C3}^{Lap} = u_{C3}^{Ref} - \frac{\frac{1}{d}}{\frac{1}{d} + \frac{1}{d} + \frac{1}{d}} (u_{F3}^{Ref} + u_{Cz}^{Ref} + u_{C3}^{Ref}) \quad (2.1)$$

$$u_{P3}^{Lap} = u_{P3}^{Ref} - \frac{\frac{1}{\sqrt{d^2+d^2}}}{\frac{1}{d} + \frac{1}{\sqrt{d^2+d^2}} + \frac{1}{d}} u_{Cz}^{Ref} - \frac{\frac{1}{d}}{\frac{1}{d} + \frac{1}{\sqrt{d^2+d^2}} + \frac{1}{d}} (u_{C3}^{Ref} + u_{Pz}^{Ref}). \quad (2.2)$$

The  $u$  refers to potential and  $d$  is the distance between the electrodes and is 6 cm in this work.

## 2.2.2 Bandpass filtering

Since the main brain activity happens between 4 and 40 Hz (see 1.2.2) frequencies above and below that limit are not necessarily needed and they can be removed. This was done by bandpass filtering the signal with FIR bandpass filter with cut-off frequencies at 4 and 40Hz. FIR is the abbreviation for *Finite Impulse Response* (*FIR*) filter and its main benefit is that it has linear phase response and it suffers less from the finite wordlengths [IEC93].

Bandpass filtering was done by lowpass/highpass filter pair of orders 8 and 20. These filters' frequency and phase responses are shown in figures (2.2) and (2.3).

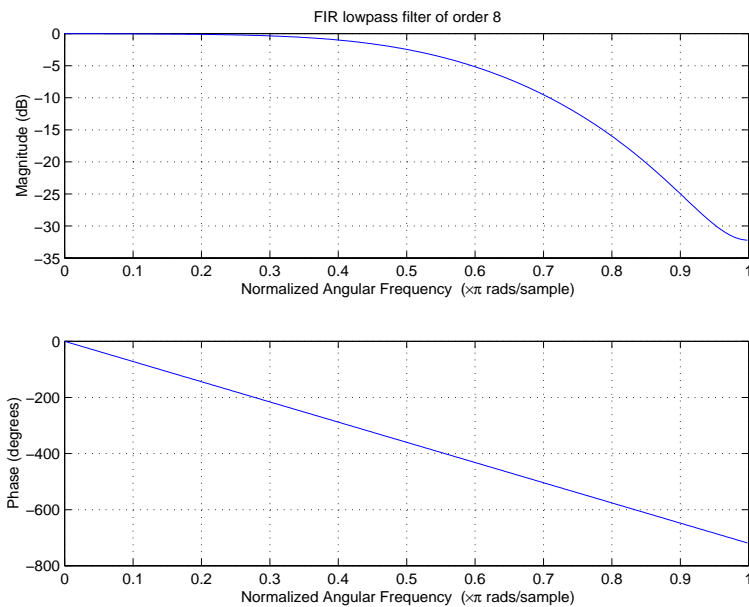


Figure 2.2: FIR lowpass filter of order 8

The effect of the bandpass filtering is shown in figure (2.4) where can be seen how bandpass filtering has removed spectral components from 0.5 second segment of EEG signal.

## 2.2.3 Wavelet denoising based on NML criterion

This denoising method removes white noise from the signal. The variance of the noise is estimated by the *NML* -criterion which helps to divide the signal into two parts: The information content and the noise content. As the wavelet decomposed signal is rebuild by adding the basis functions one at a time, the information criteria suggest the point when

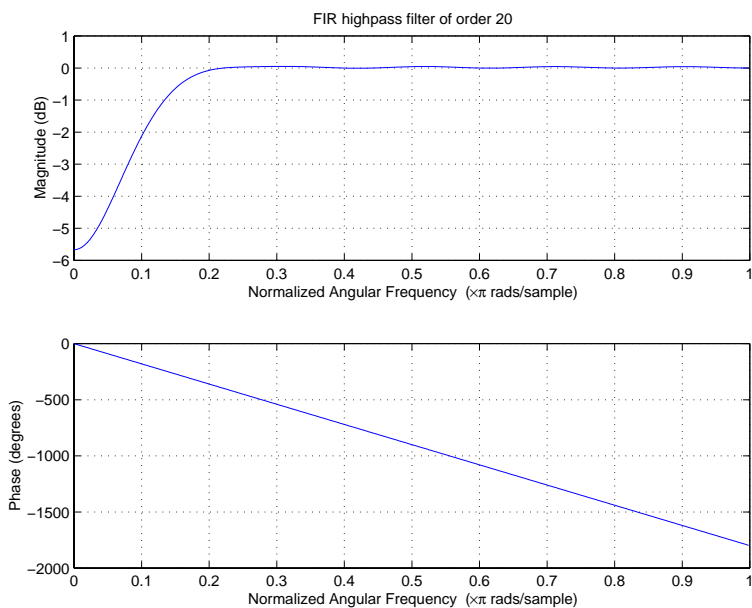


Figure 2.3: FIR highpass filter of order 20

the basis functions added do not increase the information content of the rebuild signal. The actual denoising algorithm has following steps:

1. Do the wavelet decomposition for the given data to get the wavelet coefficients  $M$
2. Sort the absolute values of wavelet coefficients to decreasing order
3. For  $k = 1$  to  $M$ , take a subset of  $k$  largest coefficients and set the rest  $M - k$  to zero
4. Calculate the NML -criterion for the current model and store it
5. After the NML -criterion has been calculated to all  $M$  coefficients, choose the subset of  $k$  largest coefficients, which had the smallest NML -criterion
6. Finally, take the subset  $k_{max}$  coefficients and set the rest to zero and rebuild the signal

First a *wavelet decomposition* is done to the signal i.e. the signal is defined as a sum of *wavelet basis functions*. This procedure is analogic to the *Fourier analysis* in which the basis function are the sine and the cosine. The difference between the wavelet and Fourier basis functions is that the wavelet basis functions are localized both in *time and frequency*, whereas the Fourier basis functions are only localized in time. This is illustrated in figure (2.5). This means that Fourier transform uses a constant sized window

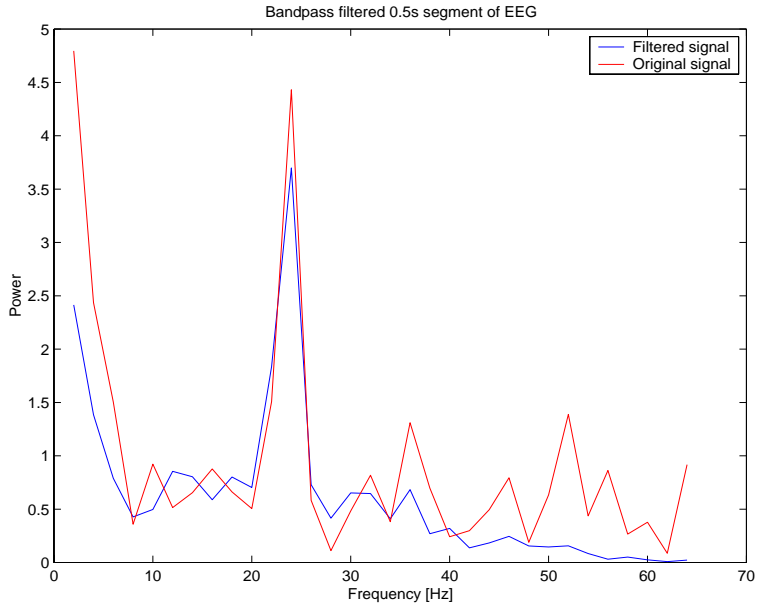


Figure 2.4: Bandpass filtered 0.5s segment of EEG

for all frequencies, therefore the resolution of the analysis being the same at all locations of the time-frequency plane. The advantage of the wavelet analysis is that the window size vary, e.g. when isolation of signal discontinuity is needed, which requires very short basis function, at the same time when detailed frequency analysis is needed, which requires long basis functions. To achieve this one must have short high-frequency and long low-frequency basis functions as shown in figure (2.5b).

In the wavelet analysis these are acquired by *scaling* and *shifting* the basis function. The wavelet coefficients  $K(a, b)$ , which are the inner product of the analyzed signal and the basis function, represent the correlation between the signal and the basis function and are calculated with the following equation in the continuous case.

$$K(a, b) = \int_{\mathbb{R}} s(t) \frac{1}{\sqrt{a}} \Psi \left( \frac{t-b}{a} \right) dt, \quad (2.3)$$

where  $s(t)$  is the analyzed signal,  $\Psi \left( \frac{t-b}{a} \right)$  is the basis function scaled with parameter  $b$  and shifted with  $a$ . In the discrete case the coefficients are calculated with the following equation

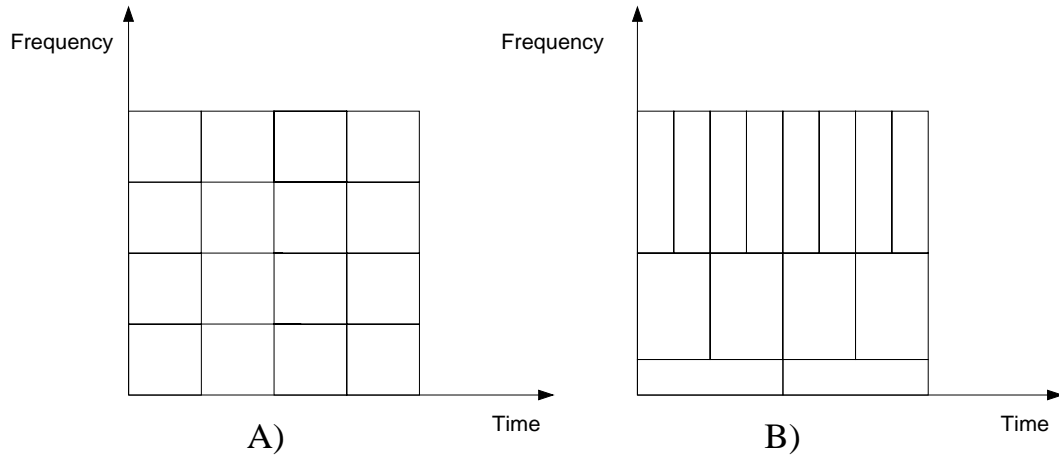


Figure 2.5: a) The Fourier transform is localized only in time. b) The wavelet transform is localized both in time and frequency.

$$K(a, b) = K(j, k) = \sum_{n \in Z} s(n) \psi_{j,k}(n) \quad (2.4)$$

where

$$a = 2^j \quad (2.5)$$

$$b = k2^j \quad (2.6)$$

$$\psi_{j,k}(n) = 2^{-j/2} \psi(2^{-j}n - k) \quad (2.7)$$

where  $a$  is the dyadic scale,  $b$  is dyadic translation,  $\psi_{j,k}(n)$  is a discrete wavelet basis and  $j \in N, k \in Z$ . Scaling is simply 'stretching' or compressing the basis function as shown in figure (2.6). The low-scale basis function catches the rapid changes in the signal, therefore correlating with high frequencies and oppositely the high-scale basis function correspond with low frequencies.

The shifting is delaying the current scaled basis function in time, producing the time localization capability as shown in figure (2.7).

The order of a wavelet is the order of its high-pass/low-pass filter pair and it tells the steepness of the filter's amplitude response at the cut-off frequency. The order of the wavelet is important depending on the application. In general the higher order filters have better frequency localization giving better bandwidth resolution and energy compaction, being therefore suitable in spectral estimation. The lower order filters have better time local-

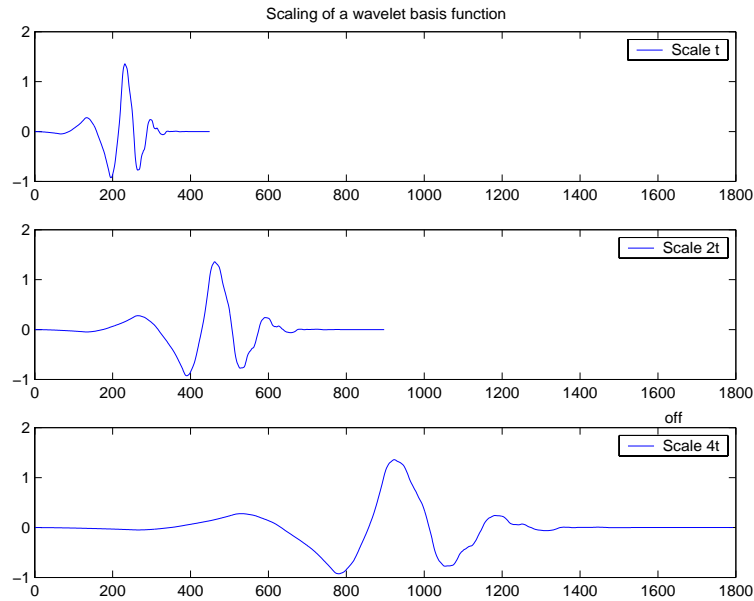


Figure 2.6: Scaling of a wavelet basis function

ization and therefore preserve better the edge information, making them more suitable in denoising, as shown in figure 2.8. For more information about the wavelets, a good book is [Mal98].

What is an *information criterion*? When choosing or comparing models, the target is to find a model that describes and predicts the underlying process that has generated the given data. In order to achieve balance between the accuracy of the model and its generalization capacity, a suitable number of parameters has to be found for the models. The information criterion is a tool to find that number of parameters. It gives a numerical value of 'goodness' of the model, making it possible to compare different models.

The *Normalized Maximum Likelihood* criterion is based on the *Minimum description Length (MDL)* by Rissanen [Ris99]. This criterion uses the *stochastic complexity* as a measure of the goodness of the model. The stochastic complexity means the shortest possible size of a code or an algorithm, which could have produced the given signal. The NML -criteria consists of two parts: The first one is the code length  $\log p(\mathbf{x}^n | \hat{\theta})$ , where  $\hat{\theta} = (\hat{\theta}_1, \dots, \hat{\theta}_n)$  are the parameters estimated from the data  $\mathbf{x}^n = (x_1, \dots, x_n)$  with *maximum likelihood (ML)* method. The second one is the code length of the ML -parameters  $\log p(\hat{\theta})$ . The problem is that the distribution of the parameters is unknown. An easy solution for this is to normalize the total code length with respect to the second part [Ris99].

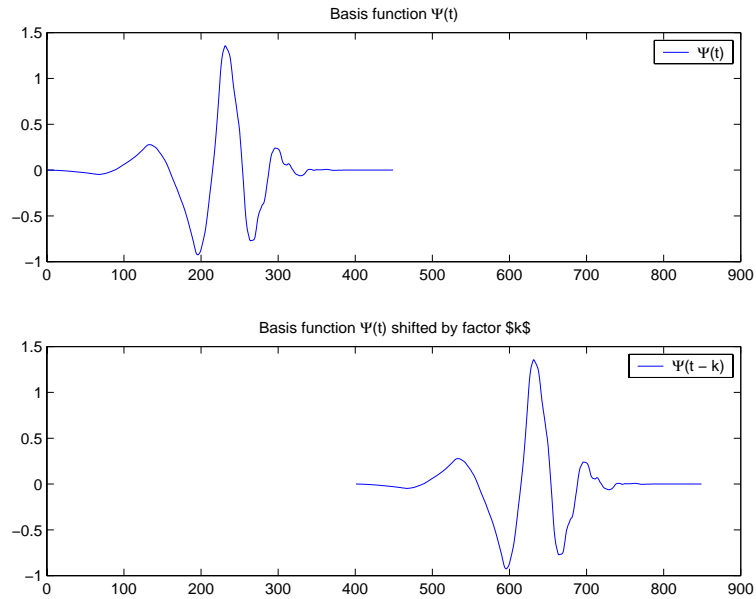


Figure 2.7: Shifting of the basis function with parameter  $a = k$

The NML -criterion works as follows: A criterion score  $I$  is calculated with

$$I = -\log p(\mathbf{x}^n | \hat{\theta}) + \log p(\hat{\theta}) \quad (2.8)$$

every time when a parameter is added to the model. As the numbers of parameters in the model increases, the performance of the model improves but its complexity and code length grows, which decreases the criterion score. The second term, the code length of the parameters, increases as more parameters are added to the model and some point when the additional parameters only provide marginal improvement to the model, the second term in (2.8) is growing faster than the first term is decreasing. This starts to increase the criterion value. The point when the criterion score stops decreasing is the point when no more parameter should be added to the model.

The actual NML -criterion when the error of the model is assumed to be Gaussian with zero mean is [Ris99]

$$\min_k \left\{ (n - k) \ln(\sigma^2) + k \ln(n\mathbf{R}) + (n - k - 1) \ln \left( \frac{n}{n - k} \right) - (k + 1) \ln(k) \right\}, \quad (2.9)$$

where  $\sigma^2$  is model's variance,  $\mathbf{R}$  is the noise variance,  $n$  is maximum number of parameters and  $k$  is a parameter  $1 < k < n$ .

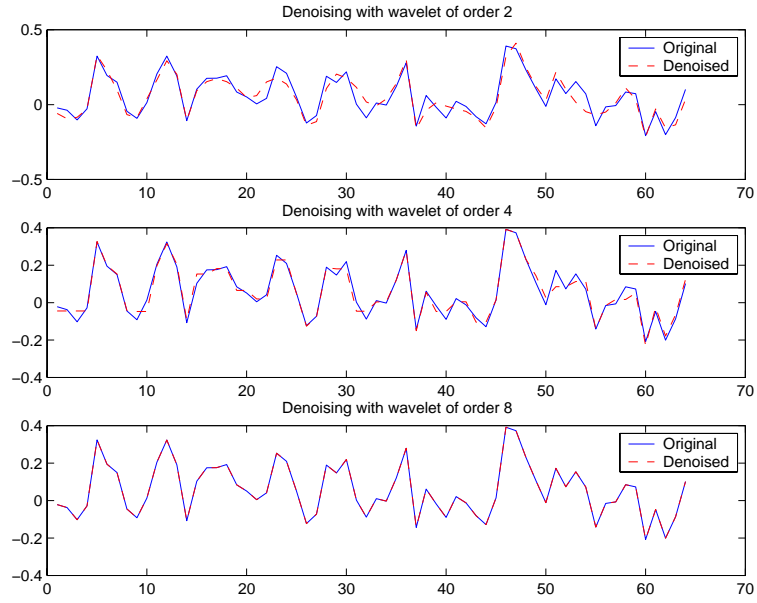


Figure 2.8: The effect of the filter order to denoising

In the wavelet denoising the parameter are the wavelet coefficients and in order to calculate  $\sigma^2$  and  $\mathbf{R}$  the model has to be rebuild every time, which is time consuming. This can be avoided if we notice that first, the wavelet transform is *orthogonal*, i.e.

$$\mathbf{x} = \mathbf{W}\mathbf{c} \quad (2.10)$$

$$\mathbf{c} = \mathbf{W}^T\mathbf{x} \quad (2.11)$$

where  $\mathbf{x}$  is the signal,  $\mathbf{c}$  are the coefficients and  $\mathbf{W}$  wavelet transformation matrix. Because of the orthonormality, the *Parseval's* equation

$$\mathbf{c}^T\mathbf{c} = \sum_i c_i^2 = \mathbf{x}^T\mathbf{x} = \sum_i x_i^2. \quad (2.12)$$

holds.

With this knowledge the equation (2.9) can be written in the form

$$\min_k \left\{ (n-k) \ln \frac{\mathbf{c}^T\mathbf{c} - \hat{S}_k}{n-k} + k \ln \frac{\hat{S}_k}{k} - \ln \frac{k}{n-k} \right\}, \quad (2.13)$$

where  $\hat{S}_k = \hat{\mathbf{c}}^T\hat{\mathbf{c}} = \hat{\mathbf{x}}^T\hat{\mathbf{x}}$  corresponds to a product of  $k$  coefficients.

## 2.3 Feature extraction

The purpose of the feature extraction is to transform the information from an EEG device to more meaningful form for the classifier. Once again the crucial component here is the ability to distinct between the important and non-important part of the data. In the EEG signal analysis the *power spectra* have been the key featurese for a long time. A power spectrum tells the frequency content of the signal, so different EEG -rhythms and frequencies are easy to point out. Probably the most used spectrum estimation is the *Fourier Transform*. The Fourier spectral features were extracted in two different ways: First was the Welch -method and second one was Blackman-Tukey -method.

Another approach is to use the transfer function of an *Autoregressive* model, which has a benefit that it includes a noise model.

The *Wavelet Transform* is gaining lots of popularity these days due to its capability to localize both in time and frequency, but in spectral estimation it does not help. Anyway the different bandwidths provided by the scaling of the basis function gives a spectrum estimate.

### 2.3.1 Fourier spectral features

These Fourier spectral features are estimated power spectra densities and they are calculated by taking the square of the modulus of the Fourier transform of the signal [IEC93]

$$|F(\omega)|^2 = \left| \int_{-\infty}^{\infty} f(t)e^{-j\omega t} dt \right|^2. \quad (2.14)$$

Fourier transform is the sum over the time of the signal  $f(t)$  multiplied real and imaginary sinusoidal components at given frequency. Before power spectral density is calculated, certain pitfalls are to be avoided. The signal samples should be length of  $2^m$  in order to avoid *scalping loss* [IEC93], which occur when a signal component falls between two harmonic frequencies and its energy is shared between them. In this case signal lengths are 64 and 256 so they fulfil the  $2^m$  requirement.

All trends should be removed beforehand because error terms caused by trends will be integrated and their cumulative effect will cause large errors in the estimated spectrum. In this case the trends are removed by fitting a first order polynomial into each window and

then subtracting the fit from the data.

Sampling the data at  $T_s$  intervals corresponds to multiplying (windowing) the signal with rectangular pulse of width  $T_s$  and height 1. Because time domain multiplying is equivalent to a convolution in frequency domain the achieved spectrum is true spectrum convolved with the rectangular window. Due to the form of the FFT of the rectangular window this causes the spurious peaks to the spectrum as the addition and subtraction of main lobes and side lobes is done at each frequency component. This is known as *spectral leakage* [IEC93]. It can be avoided by multiplying the signal with certain window, which reduces the side lobes. Such a window should have value 1 at the mid-data point and reduce to zero at each end of the window.

Drawback of using a window to suppress side lobes is that the main lobe width is increased and it spreads into the adjacent side lobes, causing aliasing. As this happens at each harmonic frequency, it aliases whole signal spectrum. This is known as *smearing*. Therefore window and its parameters should be carefully chosen in order to strike balance between frequency resolution and spectral estimate.

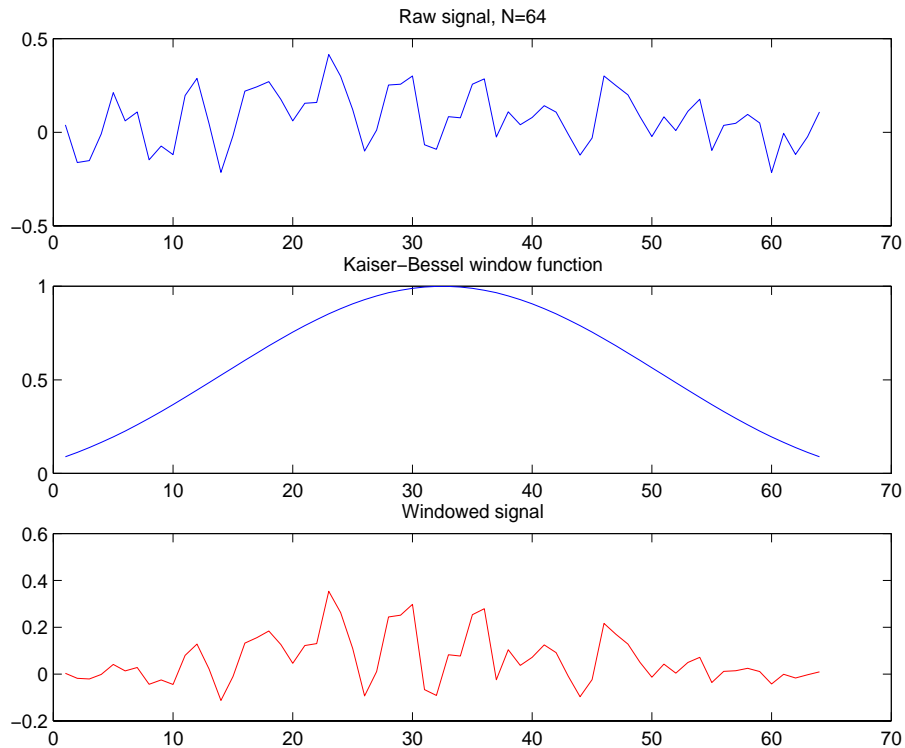


Figure 2.9: Effect of the windowing to a raw signal

The window for preventing leakage used in this work was Kaiser-Bessel window as it was mentioned to be one of the most effective one [IEC93]. Kaiser-Bessel window is a prolate-spheroidal window, for which the main lobe energy to side lobe energy ratio is maximized. The Kaiser -Bessel window is calculated by

$$w(n_{KB}) = \frac{I_0 \left\{ \pi \alpha \left[ 1.0 - \left( \frac{n_{KB}}{N/2} \right)^2 \right]^{1/2} \right\}}{I_0(\pi \alpha)}, \quad (2.15)$$

where  $0 \leq |n_{KB}| \leq N/2$ ,  $w(n_{KB})$  is the window function sample number,  $\alpha$  is a parameter which may be adjusted to select the best main lobe width to side lobe width ratio and  $N$  is the number of sample points in a window.

The actual power spectrum estimate was calculated using two different methods: The Welch method and the Blackman-Tukey method.

The Welch method is a modified *periodogram* method where the inconsistency of the periodogram method, that the increasing the number of samples does not reduce variance, is downgraded by averaging overlapping consecutive samples of the signal.

In Blackman-Tukey method the power spectrum of the signal is calculated from the discrete Fourier transform of the autocorrelation function of the data [IEC93]. Here the smoothing effect achieved from the autocorrelation function rather than from the averaged periodograms, therefore having better spatial resolution. In this method windowing has extra importance because at larger lags fewer data points are available for computation so those estimates are less accurate. Windowing emphasizes shorter lags, thus giving them greater weight when calculating spectrum estimate. So the whole procedure was:

- Calculate autocorrelate function of the data
- Multiply it by suitable window (here Kaiser-Bessel)
- Compute the FFT of the resulting data without squaring it

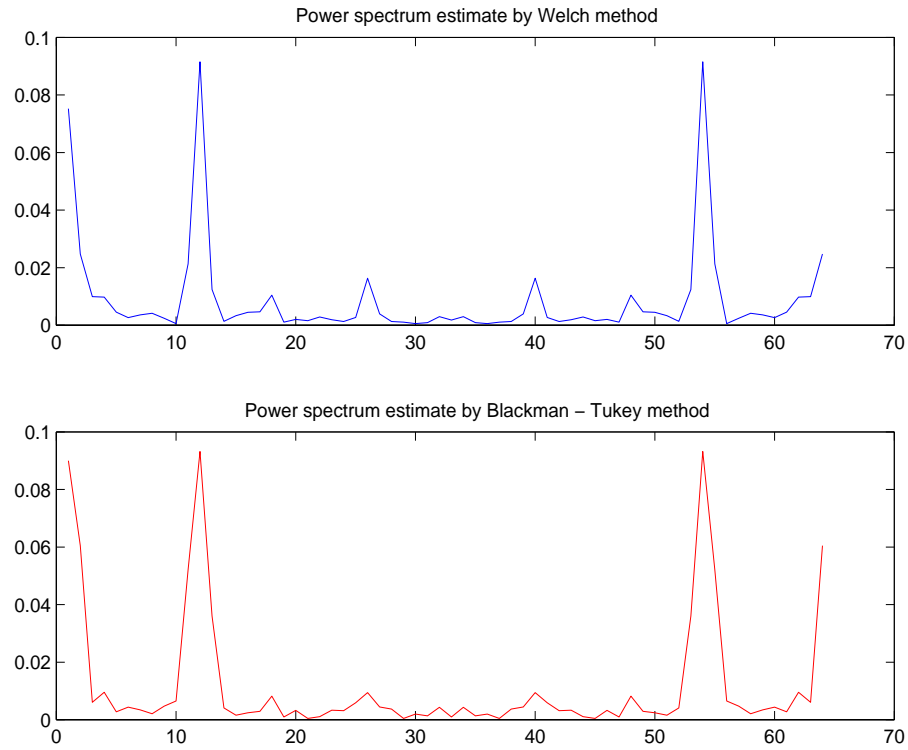


Figure 2.10: Power spectras extracted with two different method

### 2.3.2 Autoregressive spectral features

The autoregressive features were calculated from the transfer function of an *autoregression model*, in which a present value  $x_n$  or future values  $x_{n+i}$ , ( $i = 1, 2, \dots$ ) are estimated by using the previous values  $\{x_{n-m}, \dots, x_{n-1}\}$  [The92].

An autoregressive model consists of weighted sum of predefined amount  $m$  of previous values and a noise component, which is assumed to be white noise with variance  $\sigma^2$ .

$$x_n \sim N(\mu, \sigma^2) \quad (2.16)$$

$$\mu = \sum_{i=1}^m a_i x_{n-i} \quad (2.17)$$

where the coefficients  $\mathbf{a} = \{a_1, \dots, a_m\}$  are the model's parameters and  $m$  is the order of the model.

The parameters can be solved e.g. by *pseudoinverse* solution [Bis95]

$$\begin{aligned}\mathbf{a} &= (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{t} \\ \sigma^2 &= \frac{1}{n} (\mathbf{t} - \mathbf{A}\mathbf{a})^T (\mathbf{t} - \mathbf{A}\mathbf{a}),\end{aligned}\tag{2.18}$$

where  $\mathbf{t}$  includes all the present values to be estimated and  $\mathbf{A}$  are the corresponding previous values.

The spectral features are formed by calculating the frequency response of the transfer function of the AR -model. The transfer function is an all-pole IIR filter of form

$$H(z) = \frac{1}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_m z^{-m}}\tag{2.19}$$

where  $a_1, \dots, a_m$  are the coefficients of the AR -model and  $m$  is the order of the model. The frequency response is then given by

$$|H(e^{j\omega T})| = \frac{1}{1 + a_1 e^{j\omega T} + a_2 e^{2j\omega T} + \dots + a_m e^{mj\omega T}}\tag{2.20}$$

The benefit of these spectral features compared to the Fourier features is that the AR -features are limited by the order of the model i.e. the AR -model gives a generalized representation of the process and therefore it can not include all the noise in the signal.

### 2.3.3 Wavelet spectral features

As the wavelet coefficients (see 2.2.3) represent a response given by a filter of a certain bandwidth (dependent of the *scale* used), a squared coefficient gives the relative energy in that frequency band. Then by combining these coefficients a power spectrum can be formed. Naturally the localized information given by *shifting* gives no additional benefit.

### 2.3.4 Calculating the Features

The features are formed in a following way:

- Calculate the power spectrum with a selected method for every channel used in the measurement.

- Pick desired frequency bands from the spectra, e.g.  $\alpha$  and  $\beta$  bands and combine them all into a one feature.
- Finally, move to decibel scale by taking the logarithm.

## 2.4 Classifiers

In this work following classifiers were used to separate the three given classes: linear classifier Radial Basis Function (RBF) -net and a non-linear model Multi Layer Perceptron (MLP) committee Both of these classifiers provide techniques for approximating arbitrary non-linear functional mappings between multidimensional spaces. In both cases the mapping is provided as a sum of parametrized functions of a single variable. However, the particular structure of the networks is quite different so they deserve some comparison.

The MLP classifier divides the input space distinct classes by using hyperplanes, whereas RBF fits a hypersphere over a class. More accurately the activation of a hidden unit, which is weighted linear sum of its inputs and transformed by monotonic activation function, is constant on parallel surfaces of a  $(d-1)$ -dimensional hyperplanes in a  $d$ -dimensional input space. The hidden units of a RBF classifier, by comparison, use a distance to the center of the basis function after a transformation by a usually localized function. Therefore the activation is constant on concentric  $(d - 1)$ -dimensional hyperspheres

MLPs are said to have a distributed representation, as many hidden neurons contribute for the output value of a given input. The output is given by weighted linear combination of the final layer, which should give a correct value for a range of possible inputs. This requires interference and cross-coupling between the hidden neurons, which makes the network training process highly non-linear. Non-linear training introduces problems such as problem with local minimas, nearly flat areas in the error function arising from the cancellations in the effect of different weights. This can lead to a very slow convergence even with the most advanced optimization methods. The output of the RBF classifiers are typically affected only by a few hidden units, because localized basis functions of the RBF classifier form a representation in the hidden unit space, which is local respect with the input space.

Training of a MLP classifier is a supervised training, where all the parameters of the classifier are determined simultaneously as a part of single global training strategy. RBF classifier are trained in two parts: First part is determining parameters of the basis functions by unsupervised methods and then second layer weights are found by fast linear supervised methods.

## 2.4.1 Multi Layer Perceptron (MLP) committee with Early Stopping regularisation

MLP -net is nowadays very popular non-linear method for classification and regression, which consists of layers (basically allways two) of its basic units, the *perceptrons*. The complexity of a MLP -classifier is controlled by the number of the perceptrons and the latent variables, *weights*, assigned for the perceptrons. Output a MLP -classifier could be e.g.

$$y(\mathbf{x}_k) = \tilde{g} \left( \sum_{j=0}^M \omega_{kj}^{(2)} g \left( \sum_{i=0}^d \omega_{ji}^{(1)} x_i \right) \right) \quad (2.21)$$

where  $\sum_{i=0}^d \omega_{ji}^{(1)} x_i$  corresponds to the outputs of the first layers (called a *hidden layer*)  $j$  perceptron for the  $i$  part input.  $g()$  is a *transfer function* and  $\sum_{j=0}^M \omega_{kj}^{(2)}$  is the output of the output layer. In an output layer is an output unit corresponding to the each  $k$  target class. For the basic principles of a MLP reader is suggested to see [Bis95]. Here only some key issues are dealt with such as architecture, regularization and weight optimization.

Used architecture in this MLP is a typical two layer net with ten hidden neurons and three output neurons corresponding to the three different classes. Used transfer functions were *tansig*, i.e. the tanh function

$$g(a) = \frac{e^a - e^{-a}}{e^a + e^{-a}} \quad (2.22)$$

for the hidden layer and *softmax*

$$\tilde{g}(a) = \frac{e^{a_k}}{\sum_{k'} e^{a_{k'}}} \quad (2.23)$$

for output layer.

The regularization method used was *Early Stopping*. But why is regularization so important? Key issue of any classifier is its *generalization* capacity. The model should not learn the data but the *process* that generates the data. Therefore great care should be taken when determining the complexity of the model as too complex model will only learn the given data without being able to replicate the process and too simple model has a high systematic error in it.

Early Stopping is an effective method to control the complexity of a MLP- net [Bis95].

In this method training process is stopped when error of a *validation set* which is *independent* from the *training set* starts to increase. This can be rationalized as follows. At the beginning of the training weights of the neurons are small i.e. the mapping is almost linear. The training of the MLP net is a iterative process where training error decreases monotonically, which increases the effective complexity of the net. Correct complexity can be achieved by stopping the training when error of the *validation set* starts to increase, because at that point the *generalization* capacity of the net is at its peak. If the training is continued, the still increasing complexity will specialize the net to given data.

A drawback of this method is that it is very sensitive to initial weight values as different initial weight values may lead to different local minima. A solution for this is to train a *committee* of MLP -nets [Bis95] and calculate the average of the results. In this work a *committee* of 10 MLP -nets was trained and their results averaged.

For the weight optimization, i.e. finding the latent variables that weight the perceptrons, two algorithms were used. The first one is an algorithm that uses *global* knowledge of the state of the overall weight-update vector and is called *scaled conjugate gradient (SCG)* [Bis95]. The second method is based on weight specific information only, such as the temporal behaviour of the partial derivate of this weight. These type of methods are called *local* adaptive methods. Such a method used here is *Resilient Propagation (Rprop)* [Rie94].

Rprop is a local adaptive learning method. In this method the influence on weight step is not determined by the size of the partial derivatives of the error function but the *sign* of it. The actual weight step is defined by a constant, weight specific *update-value*.

The benefits of the Rprop is that it is not computationally expensive and the adaptation is not blurred by unforeseeable gradient behaviour, which means that large values of partial derivatives respect to the error function do not 'throw' the search to completely new areas on the error surface.

On the other hand the drawback are that the weight update can be calculated only after the sum of all errors is calculated. This can reduce the efficiency on redundant training sets and may cause problems with variable training sets. The Rprop is described in more detail in appendix A.1

SCG is based on the gradient search, where at the each round of iteration the minimum is searched along line given by negative gradient of the error function at the current position. This error minimization proceeds usually very slowly as each new gradient is orthogonal to previous line search direction. One solution to this problem is to choose new search

directions so, that at each step of the iteration the component of the new search direction parallel to the old search direction remains zero (to lowest order). It can be shown that if the search directions are selected in this way, after steps equal to the dimension of the search space have been taken, we have arrived to the minimum of quadratic error surface. A problem of this approximation is that in a non-quadratic error function the Hessians used in the algorithm may not be positive definite, which leads to the increase of the error in the iteration step. This problem can be overcome by modifying the Hessian to ensure that it is always positive definite. This is done by adding some multiple of unit matrix to the Hessian i.e. *scaling* the Hessian. The SCG is one of the most effective optimization algorithms around and used extensively with neural networks [Bis95]. The SCG is described in more detail in appendix A.2.

## 2.4.2 Radial Basis Function (RBF) nets

Characteristical to the RBF models is that everything is tried to keep as linear as possible. This will keep calculation simple (only linear algebra is needed) and also the processing time required is relatively low. A RBF -net classifies the input with a linear equation

$$y(\mathbf{x}, \omega) = \sum_{j=1}^m \omega_j h_j(\mathbf{x}), \quad (2.24)$$

where the  $\mathbf{x}$  is the input vector,  $\omega$  are the weights which are solved linearly e.g. with the pseudoinverse solution, and  $h_j(\mathbf{x})$  is the *basis function*. For the basic principles of the RBF classifiers reader is suggested to see [Orr96] or [Bis95]. The complexity was determined by algorithm called *forward selection*, which was then tuned by *local ridge regression*.

Used basis function were popular gaussian function

$$h(x) = \exp\left(-\frac{(x - c)^2}{r^2}\right), \quad (2.25)$$

where  $c$  is the centre and  $r^2 = \sigma^2$  is the radius of the of the function. Gaussian basis function are local i.e. they give significant response only in a neighbourhood of the centre and they are radial, which means that their response decreases monotonically with distance from the center.

The initial kernel widths were acquired by *10-fold cross validation* procedure. The initial training data, in this case the first session data (see section 2.1, was divided to ten parts. Then the model was trained with nine parts and tested with the tenth part. This procedure was repeated until all parts had been used as a test part. The model was trained with 20 different kernel widths, ranging logarithmically spaced from 0.1 to 10.

Idea of the forward selection is to find a subset of basis functions drawn from fixed set of candidates. There is  $2^M - 1$  subsets in a set of size  $M$ , so heuristics must be used to find a small but sufficiently good fraction from the space of all subsets. This algorithm starts from the empty subset to which one subset which most reduces SSE is added at a time until certain criterion is met. In this work the criterion used was *generalized cross-validation (GCV)*.

Forward selection was used mainly to find the suitable number of hidden units (basis function) for the first layer. These basis functions had fixed values for their centres and radius. Now *local ridge regression*, also known as *regularisation*, is used to fine tune the centers and radius to improve the performance of the classifier. Regularisation reduces the complexity of the classifier. Reduction of complexity is done by reducing effective number of parameters. Resulting loss of flexibility makes the model less sensitive. Here is how regularisation works. Let us define a cost function

$$S = \sum_{i=1}^p (t_i - y(\mathbf{x}_i))^2 + \lambda \sum_{j=1}^m \omega_j^2, \quad (2.26)$$

where  $t_i$  is target,  $y(\mathbf{x}_i)$  is corresponding output of the model,  $\lambda$  regularisation parameter and  $\sum_{j=1}^m \omega_j^2$  is a squared sum of the weights. So in regularisation a *penalty term* is added to SSE, consisting a squared sum of the weights. Effect of this is that the large weights are penalized more. The gradient of (2.26) respect to  $\omega$  is  $\omega$ , so the cost function exponentially reduces the weights towards the zero. This bias introduced by the regularisation then favors small weights, which smoothes the output function, since large weight values are usually needed for producing highly variable output. The effect of the regularisation parameter  $\lambda$  is such, that large values of  $\lambda$  means that tight fit is sacrificed if it requires large weight values. On a contrary small  $\lambda$  makes a rather tight fit possible.

If we use only one value of  $\lambda$  to each basis function, then we will have problems if these functions have significantly different smoothness in different parts of the input space. Natural answer to this problem is to set an individual value of  $\lambda$  to each basis function. Each basis function can be modified individually by setting the basic cost function (2.27)

to form

$$S = \sum_{i=1}^p (t_i - y(\mathbf{x}_i))^2 + \sum_{j=1}^m \lambda_j \omega_j^2. \quad (2.27)$$

In general there is nothing local in doing this kind of regularisation, but if we use local basis functions, such as gaussian, the the smoothness is controlled in a local fashion by the individual regression parameters. Regression parameter  $\lambda$  is select using GCV criterion and the equation for projection matrix in equation (A.17) is

$$\mathbf{P} = \mathbf{P}_j - \frac{\mathbf{P}_j \mathbf{h}_j \mathbf{h}_j^T \mathbf{P}_j}{\lambda + \mathbf{h}_j^T \mathbf{P}_j \mathbf{h}_j}, \quad (2.28)$$

where  $\mathbf{P}_j$  is the projection matrix after the  $j$ th basis function has been removed and  $\mathbf{h}_j$  is the  $j$ th column of the design matrix. Optimization is done one  $\lambda_j$  at a time and then repeating until they all converge. Because the basic network complexity is already defined with the forward selection, starting  $\lambda$  are set to

$$\lambda_j = \begin{cases} 0 & \text{if } j \in \mathbf{S} \\ \infty & \text{otherwise} \end{cases} \quad (2.29)$$

where  $\mathbf{S}$  is the subset of basis functions given by the forward selection method.

# Chapter 3

## Experiments

Four different experiments were conducted. The first one studied the preprocessing, the second one compared the feature extraction methods based on the Fourier transform and the third one the other feature extraction methods and the fourth the effect of the weight optimization algorithm with MLP model.

### 3.1 Definitions

The success of a classification is measured by three criteria: The first one is the *correct classification rate*. This means quite self explanatory that we want all the samples to be classified correctly. The classification results are given in *confusion matrix*, as in the example in figure (3.1). The rows of the confusion matrix correspond to the correct class and the columns to the estimated class. In the diagonal of the confusion matrix the classification rate of a class can be found. In the example above class 1 was classified with 87.9% success, class two 69.9% and class three 32.3%. The rows tell the error classifications, e.g. 12.1% of class 1 was classified as class 2.

<b>Day 1.</b>	Class 1	Class 2	Class 3
Class 1	87.9	12.1	0.0
Class 2	13.5	69.9	16.6
Class 3	26.8	40.9	32.3

Table 3.1: An example of a confusion matrix

Since the misclassifications are especially harmful in the system's applications, the another criterion is the amount of misclassifications, also called *the number of false positives*. This amount can be found from the confusion matrix by examining the values that are not in the diagonal. In the example above e.g. 40.1% of the classification of the class 2 were wrong (13.5 + 16.6). The false positive level can be reduced by *thresholding* the classification i.e. only accepting the classifications where the classification probability exceeds certain threshold. In this work the thresholds were chosen so that they give the best performance for the classifier and they were usually around 90%.

Third criterion tells the amount of information in bits that the model is able to transmit successfully from the user to the application and is called the *channel capacity*. The channel capacity is calculated by maximizing the *mutual information* between the output and the input respect to the input. The mutual information tells how the uncertainty of the input reduces when we know the output (and vice versa). The channel capacity is explained in detail in Appendix B.

## 3.2 Training and Testing Procedure

All the four described models were used in the experiment, except experiment four, where only the MLP was used, since its purpose was to find out the effect of the different weight optimizations.

All the models used in the experiments were trained using the same approach: The initial training was executed with the data from the session 1 in table 2.1. This training was then tested with the session 2. Then the model was trained with the data of the session 2 and tested with session 3 data. And again following the same principle trained with session 3 data and tested with session 4 data. The models were not trained with session 4 data or tested with the session 1 data. By following this procedure as high statistical independence as possible between the training and test set was acquired (they were recorded in different days). Also this gives insight how the user's skill evolves.

### **3.3 Experiment 1.**

In this experiment the effect of the different preprocessing methods for the classification were studied. The used feature extraction method were Welch periodograms (see 2.3.1). The experiment begun with testing the models without any preprocessing to get a kind of a 'baseline' of performance. Then the Laplacian filtering was applied and the models were tested. Then bandpass filtering was added and finally wavelet denoising. Both classifiers were trained ten times and the results are the average. The kernel width of the RBF classifier was searched with cross validation procedure, explained in section (2.4.2). In order to have a point of comparison for the different preprocessing methods, the outputs of the classifiers were calculated also with the used data which was preprocessed only with the basic preprocessing methods; trend removing and baselining.

### **3.4 Experiment 2.**

This experiment compared the two feature extraction methods based on Fourier transform: The Welch method and the Blackman-Tukey method. As the result from the experiment 1. showed (see section 4.1, the best preprocessing method, the Laplacian filtering was applied before extraction. The following segment parameters were used for Welch:

- Window of two 0.5s with 50% overlap
- Window of four 0.5s with 75% overlap

The used model was MLP.

### **3.5 Experiment 3.**

Here the different feature extraction methods were tested. Since the experiment 1. showed (see section 4.1) that Laplacian filtering is the only significant preprocessing method, it was only preprocessing that was applied. Again both classifiers were trained ten times and the results are the average.

## **3.6 Experiment 4.**

Here the two weight optimization algorithms were compared with MLP using Welch features and Laplacian preprocessing.

# Chapter 4

## Results

### 4.1 Experiment 1.

The key finding from the experiment 1. is that the only relevant preprocessing method is the Laplacian spatial filtering. This is obvious when examining the results shown below. The results for the MLP classifier are presented in figures (4.1) and (4.2). The results were acquired by using classification probability threshold  $\rho$  of 90%. The Laplacian improves the average false positive rate from 75% of the non-preprocessed to 25%. In the day three the false positive rate was 11%, which was the best performance of the Fourier features. This rate was acquired using probability threshold  $\rho$  of 90%.

The deterioration of performance in bandpass filtered data is most probably due to the fact that the most information bearing frequency bands, the  $\alpha$  (8-12 Hz) and the  $\beta$  (around 30 Hz) are located close to the cut-off frequencies. As can be seen from figures (2.2) and (2.3) these filters are not ideal, i.e the attenuation is not a step function, so also the frequencies close to the cut-off frequencies are also attenuated. In this case the attenuation of the  $\alpha$  and  $\beta$  band was severe enough to affect the classification. This could probably be avoided by better filter desing but that is not in the scope of this work.

Also the NML based denoising was disappointment. This is most likely due to the fact that the variance of the EEG signal is so high as noted in section (2) that the NML - criterion decides that most of the data or some important component is plain noise and it is removed. The standard deviation  $\sigma^2$  for the classes 1, 2 and 3 in the day 1 were  $2.7 \cdot 10^{-3}$ ,  $1.4 \cdot 10^{-3}$  and  $1.4 \cdot 10^{-3}$  respectively. The standard deviations for the noise ( $\mathbf{R}$ )

determined by the NML criterion were  $0.3 \cdot 10^{-3}$ ,  $0.2 \cdot 10^{-3}$  and  $0.3 \cdot 10^{-3}$ . On the average 25% of the wavelet coefficients were removed. This means that the denoising most probably removes some important frequency components because  $\mathbf{R}$  is only about 10% of total  $\sigma^2$  and the rejected amount of wavelet coefficients is relatively low. The used wavelet was bio-orthogonal family of order 3 in decomposition and order 1 in reconstruction.

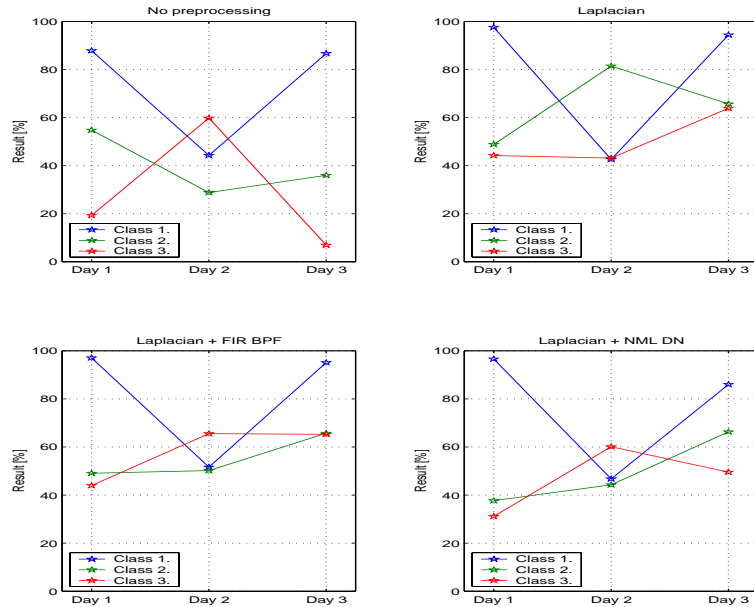


Figure 4.1: Preprocessing results for the MLP classifier with  $\rho = 90\%$

There are major differences in the daily performances. The most noticeable is the 'second day lapse', especially the class 1 took deep dive in all the preprocessing methods. This could be explained perhaps with a nervousness or such of the subject, since the class 1 corresponds to relaxation task. The channel capacity were at the best about 2 bits/s and the rejection rate was about 20% so every fifth classification was rejected.

More detailed picture about the effect of Laplacian filtering is shown in table (4.1). The unprocessed MLP is not applicable in any circumstances. The Laplacian filtering is especially crucial to distinct the classes 2 and 3. Also the amount of rejected samples is halved.

The results for the RBF classifier are shown in figures (4.3) and (4.4). The cross validation gave kernel widths between 20 and 30 for the different preprocessing methods. The RBF's performance is clearly weaker than the MLP's, except in the false positive rate. The highest usable rejection rate was only 50%. This is mainly due to the way these simple versions of RBF classifier work. The fixed sized prototypes are chosen from a large group

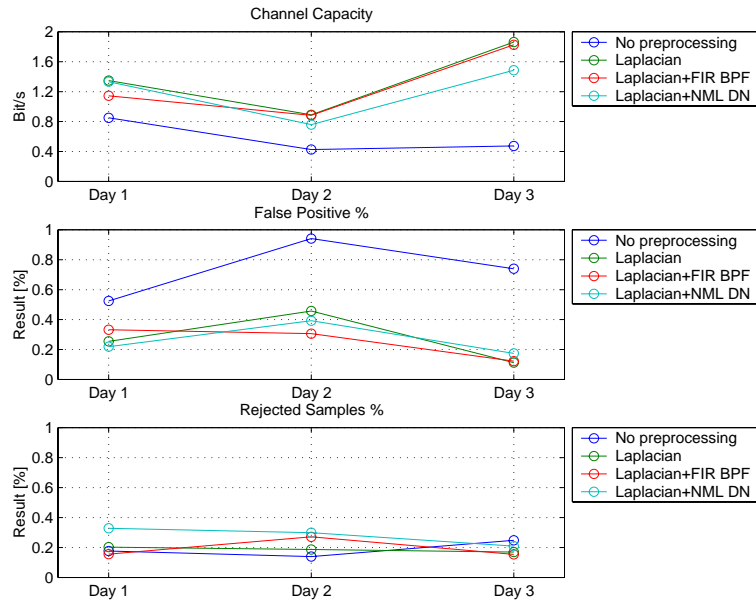


Figure 4.2: Preprocessing performances for the MLP classifier with  $\rho = 90\%$

of predefined set of prototypes of fixed size. These chosen prototypes are not selected effectively enough to form as good decision planes as the MLP. But since the basis function is radial, the probability given by the classifier outside these selected prototypes quickly decreases, keeping the false positive rate small. On the other hand, the number of rejected samples is high, which keeps the channel capacity down.

## 4.2 Experiment 2.

In this experiment different methods to form Fourier features were compared. The results are shown in figure (4.5) and in table (4.2).

The bottom line here is that there is no major difference between the 1s and 2s window used in the Welch method and the Blackman - Tukey gives inferior results when compared to the Welch method. This could have been anticipated from figure (2.8). The peaks in the feature extracted with the Blackman-Tukey are not as tall and sharp as in Welch feature. This is probably due to the high variance and noisiness of the EEG signal, which the averaging used in the Welch cancels out more effectively than the autocorrelation function used in the Blackman-Tukey method. In the table (4.2) are summarized the daily performances of the three methods.

Day 1.	Class 1	Class 2	Class 3	Rejected
Class 1	87.4	12.1	0.0	0.5
Class 2	11.4	49.4	6.5	32.7
Class 3	24.4	24.8	15.4	35.5

Day 2.	Class 1	Class 2	Class 3	Rejected
Class 1	44.3	3.3	42.6	9.8
Class 2	4.2	27.1	43.9	24.8
Class 3	4.6	17.8	56.0	21.6

Day 3.	Class 1	Class 2	Class 3	Rejected
Class 1	86.7	2.8	1.4	9.1
Class 2	23.4	26.9	2.3	47.4
Class 3	33.3	17.6	4.2	44.9

Day 1.	Class 1	Class 2	Class 3	Rejected
Class 1	97.6	0.0	1.5	1.0
Class 2	11.7	48.8	11.2	28.3
Class 3	15.0	9.3	44.2	31.5

Day 2.	Class 1	Class 2	Class 3	Rejected
Class 1	42.6	13.1	22.1	22.1
Class 2	0.0	81.5	9.5	9.0
Class 3	0.0	31.7	43.2	25.1

Day 3.	Class 1	Class 2	Class 3	Rejected
Class 1	94.4	1.4	0.7	3.5
Class 2	2.9	65.7	8.6	22.9
Class 3	3.2	8.3	63.9	24.5

Table 4.1: Left: Results from Unpreprocessed MLP. Right: Results from Laplacian filtered MLP.  $\rho = 90\%$ .

Day 1.	Channel Capacity	False positive %	Rejection Rate %
Welch: 2s	1.30	23.24	16.12
Welch: 1s	1.24	25.46	17.35
Blackman-Tukey	0.98	31.40	20.03

Day 2.	Channel Capacity	False Positive %	Rejection Rate %
Welch: 2s	0.99	27.03	19.24
Welch: 1s	0.92	33.47	20.57
Blackman-Tukey	0.55	39.06	23.85

Day 3.	Channel Capacity	False Positive %	Rejection Rate %
Welch: 2s	1.95	7.29	25.01
Welch: 1s	1.82	12.37	28.67
Blackman-Tukey	1.32	16.6	32.03

Table 4.2: Daily performances of the different Fourier features with  $\rho = 90\%$

### 4.3 Experiment 3.

The comparison of the three different features using power spectrum are shown in figures (4.6) and (4.6). When studying daily classification rates the Welch features have better performance at day 2, but lower performance level at days 1 and 3. The surprising result is that the wavelet features have the best channel capacity at each day, due to its good false positive rate. There was some doubt that can sufficient frequency resolution be obtainable with the discrete wavelet but it seems to be so.

The effect of the order of the wavelet basis function is shown in the figures (4.8) and (4.9). The effect is not big, but it is enough to make clear distinction on performance between classes 2 and 3. The results for the RBF classifier are shown in figures (4.10) and (4.11). From these results can be seen that the AR features are the only features that give meaningful performance, especially when examining the false positive rate. Actually

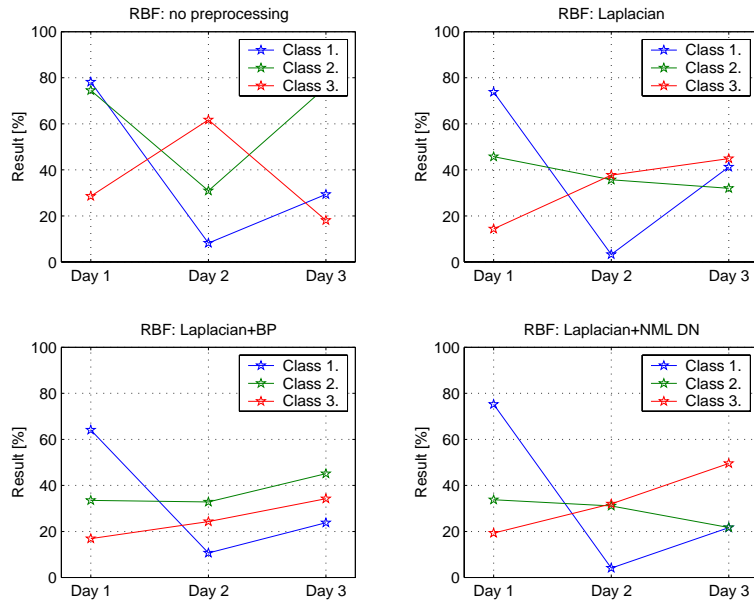


Figure 4.3: Preprocessing results for the RBF classifier with  $\rho = 50\%$

the false positive rate of the RBF classifier with AR features had overall the lowest false positive level of 5.3% in day 3. The kernel widths for the AR and the wavelet features were a bit higher than for Fourier features, being around 25 - 35.

In the tables (4.3) and (4.3) is shown the effect of the thresholding on AR features. The MLP, which uses high threshold, still has a higher rate of false positives. The MLP divides the feature space for the different classes with a discriminant plane/line made of  $M - 1$  planes/lines and this separation can be very accurate. This gives high probabilities of classification, e.g. 95% for class 1, 2% for class 2 and 3% for class 3, as the MLP model is complex in the statistical sense. But in this problem the feature space is heavily overlapped, and the samples close to the border between the classes are often misclassified, as the feature spaces of the classes overlap over the discrimination line. This combination gives false positives as the samples close to the discrimination line have high classification probability but they are often misclassified.

On the other hand, the RBF has to use low threshold, but it seems to be very effective. We also know that since the used kernel width is large (25 - 35) the probability reduces quickly as the distance to the basis function grows. Therefore only very few basis functions give significant probability. The output of the RBF classifier is a weighted sum of these probabilities, so the few larger probabilities are a kind of averaged in the summing, lowering the overall probability. Here is an example: Consider that we have 7 basis func-

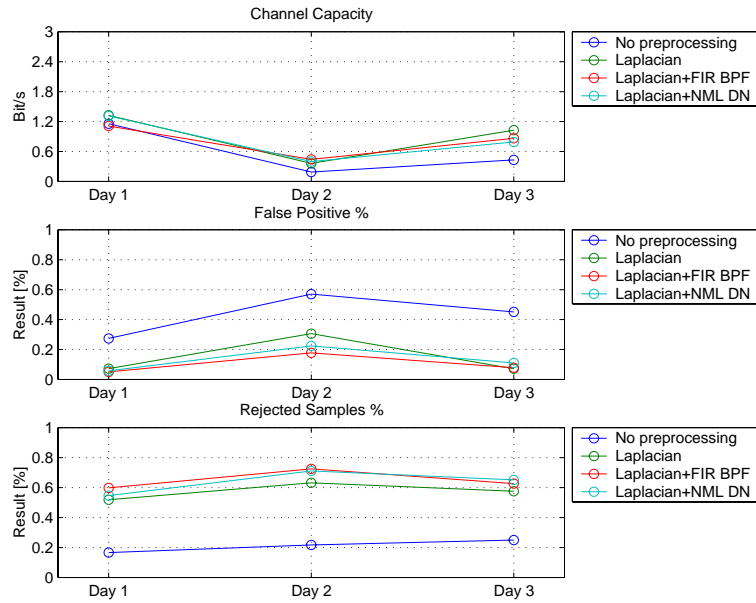


Figure 4.4: Preprocessing performances for the RBF classifier with  $\rho = 50\%$

tions and we get following probabilities out of them

Class1:	0.0457	0.1023	0.1226	0.1290	0.1745	0.1967	0.0513
Class2:	0.0284	0.0454	0.7800	0.1280	0.1350	0.1227	0.8700
Class3:	0.0290	0.1813	0.1568	0.0890	0.1232	0.1990	0.0512

There are only 2 significant probabilities and they are the third and the seventh basis functions in class 2. For simplicity the weighting is ignored. The class probabilities are

Class1:	0.1628
Class2:	0.6010
Class3:	0.2362

So even that the individual probabilities from the basis functions suggest that the sample is from class 2, the overall probability is not higher than 60%. With this kind of behaviour, even a modest threshold seems to reduce false positives effectively.

More generally this could be considered as that the basis functions of the RBF model center themselves on areas which are most populated by the samples. This way the samples outside these areas have small classification probability and are rejected by the threshold. The drawback is that in this overlapping feature space most of the samples are in this

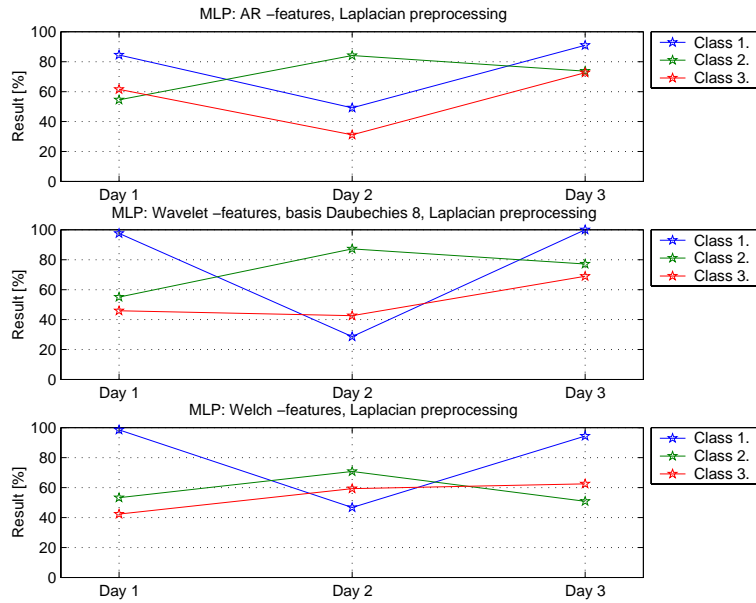


Figure 4.5: Daily results of the different Fourier features with  $\rho = 90\%$

'gray area' and therefore the number of rejected samples is higher in the RBF classifier. The more important aspect is that the false positives occur much more seldom.

## 4.4 Experiment 4.

This experiment showed that the SCG is clearly better algorithm for this model. The idea behind the testing with the RProp was that when we have a noisy signal, it may be better to have weights with small values, i.e. weights close to the origin of the weight surface [Bis95, Zha98]. This could be achieved with rather ineffective weight optimization algorithm, which will get stuck in a local minima around the origin. We want small weight values because smaller weight values produce smaller errors in equation (2.21), which may be already high with noisy signal.

But the results in figure (4.12) and in table (4.5) show that the SCG works better. This is probably due to that it finds better minima of errors which are further away from the origin of the error surface. The SCG can reach them being more effective than the Rprop.

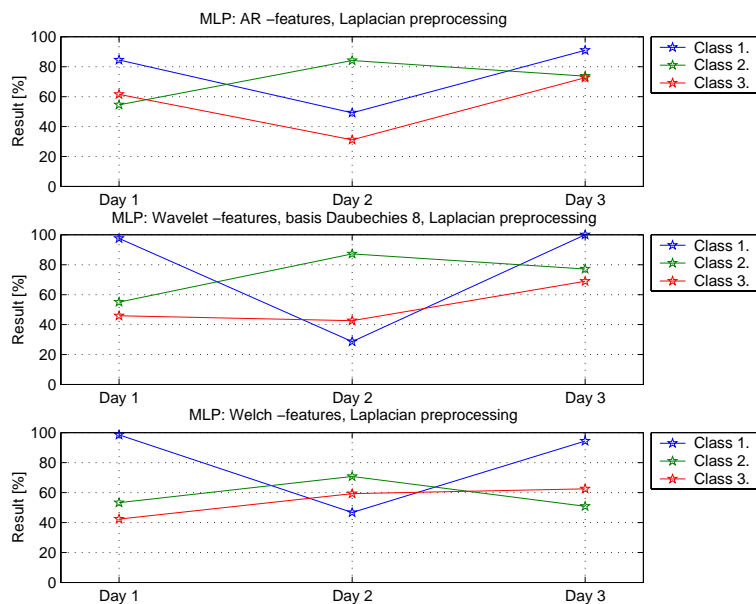


Figure 4.6: Daily results for the MLP classifier with the different features and  $\rho = 90\%$

## 4.5 Summary of the Results

In the tables (4.6), (4.7) and (4.8) are collected all the results in a compact form. In overall, the MLPs have higher channel capacity and the RBFs have smaller false positive rate. The RBFs have smaller channel capacity because they reject more samples than MLPs

From the two main criteria, the channel capacity and the false positive rate, I would place more importance for the latter one. Since the ABI is mainly intended for controlling application, false classification can not simply be tolerated, since they can have disastrous effects when e.g. controlling a wheelchair. Also the bit rates we are talking about are so small that the time required for correcting a mistake, e.g. when typing with virtual keyboard, is long.

So in that sense, the RBF classifier would provide more stable platform to build a BCI. Of course it should be remembered that this RBF classifier is very simple. An example of a more complex RBF classifier can be found on [dRM00], which is the current classifier used in the ABI.

The only certain thing about the preprocessing and feature extraction is that the Laplacian filtering is definitely essential. The different features seem to work better depending on the used classifiers so no certainty can be found. That can be said that the AR and

Day 1.	Class 1	Class 2	Class 3	Rejected
Class 1	87.9	1.9	10.2	0.0
Class 2	2.1	61.6	36.4	0.0
Class 3	6.4	22.4	71.3	0.0

Day 1.	Class 1	Class 2	Class 3	Rejected
Class 1	84.5	0.5	6.8	8.3
Class 2	0.3	54.5	30.9	14.3
Class 3	2.9	16.4	61.5	19.2

Day 2.	Class 1	Class 2	Class 3	Rejected
Class 1	54.9	26.2	18.9	0.0
Class 2	1.7	89.5	8.8	0.0
Class 3	6.6	52.2	41.3	0.0

Day 2.	Class 1	Class 2	Class 3	Rejected
Class 1	49.2	22.1	10.7	18.0
Class 2	0.6	84.2	3.8	11.5
Class 3	3.3	44.3	31.1	21.3

Day 3.	Class 1	Class 2	Class 3	Rejected
Class 1	94.4	0.0	5.6	0.0
Class 2	2.3	78.3	19.4	0.0
Class 3	11.1	2.3	86.6	0.0

Day 3.	Class 1	Class 2	Class 3	Rejected
Class 1	90.9	0.0	2.1	7.0
Class 2	1.1	73.7	12.6	12.6
Class 3	4.2	1.4	72.7	21.8

Table 4.3: Left: Results from MLP+AR features with  $\rho = 0$ . Right: Results from MLP+AR with  $\rho = 90$ .

Day 1.	Class 1	Class 2	Class 3	Rejected
Class 1	77.7	8.3	14.1	0.0
Class 2	0.0	67.3	32.7	0.0
Class 3	1.8	27.8	70.4	0.0

Day 1.	Class 1	Class 2	Class 3	Rejected
Class 1	56.3	1.5	4.4	37.9
Class 2	0.0	36.4	7.8	55.8
Class 3	0.1	14.4	48.5	37.0

Day 2.	Class 1	Class 2	Class 3	Rejected
Class 1	32.0	41.8	26.2	0.0
Class 2	0.0	92.0	8.0	0.0
Class 3	1.4	51.6	47.0	0.0

Day 2.	Class 1	Class 2	Class 3	Rejected
Class 1	15.6	8.2	9.0	67.2
Class 2	0.0	64.1	1.1	34.7
Class 3	0.0	9.3	7.7	83.1

Day 3.	Class 1	Class 2	Class 3	Rejected
Class 1	86.0	4.9	9.1	0.0
Class 2	0.0	80.6	19.4	0.0
Class 3	0.5	5.1	94.4	0.0

Day 3.	Class 1	Class 2	Class 3	Rejected
Class 1	57.3	2.1	1.4	39.2
Class 2	0.0	38.3	0.6	61.1
Class 3	0.0	0.0	53.7	46.3

Table 4.4: Left: Results from RBF+AR features with  $\rho = 0$ . Right: Results from RBF+AR with  $\rho = 50$ .

wavelet features seem to work a bit better than the Fourier, especially with RBF classifier. But it should be remembered that these features should be extracted on-line and the AR and wavelet features are computationally much more heavier to extract than the Fourier features.

But to make a final conclusions about the performance of the different classifiers, preprocessing and feature extraction methods, much deeper study should be done. This work had data from only one subject, who is rather experienced user. This fact tells that we don't have any knowledge how these methods would work on a newcomer.

Also the used data set could have had more samples. Since the feature space has large dimension  $d = 102$  and the required number of samples grows exponentially with the dimension of the data, more samples would give more accurate results. Of course with  $d = 102$  we would need senseless amount of samples. But if we keep in mind the two very fundamental assumptions of statistical modelling, that the variables of the input data is usually correlated in some way and the value of the output variables do not vary arbitrarily

<b>Day 1.</b>	Channel Capacity	False positive %	Rejection Rate %
SCG	1.33	22.64	16.12
Rprop	0.99	32.46	19.35
<b>Day 2.</b>	Channel Capacity	False Positive %	Rejection Rate %
SCG	0.99	27.03	19.24
Rprop	0.71	40.47	23.29
<b>Day 3.</b>	Channel Capacity	False Positive %	Rejection Rate %
SCG	1.95	7.29	25.01
Rprop	1.20	15.31	33.93

Table 4.5: Daily performances of the weight optimization algorithms with  $\rho = 90\%$

<b>Day 1.</b>	Class 1.	Class 2.	Class 3.	CC [bit/s]	FPR [%]	RS [%]
MLP: No prep.	87	39	8	0.83	34	33
MLP: Lap.(F)	99	53	42	1.30	23	16
MLP: Lap.+BP(F)	97	49	44	1.14	25	16
MLP: Lap.+NML(F)	97	38	31	1.33	18	33
MLP: Lap.(AR)	81	51	62	1.22	24	15
MLP: Lap.(Wav)	96	55	46	1.77	11	25
RBF: No prep.	78	75	29	1.16	28	17
RBF: Lap.(F)	74	46	14	1.32	7	52
RBF: Lap.+BP(F)	64	34	17	1.11	5	60
RBF: Lap.+NML(F)	75	34	19	1.31	6	55
RBF: Lap.(AR)	56	36	49	1.08	17	44
RBF: Lap.(Wav)	62	45	32	1.20	19	42

Table 4.6: Collected results for the day 1.

from one region to the another, but rather smoothly as a function of the input variables. The first assumption means that the input data do not fill the entire input space but rather a lower dimension sub-space. Then applying the second assumption we can do a kind of interpolation for the intermediate points between these sub-spaces. With these assumption we can therefore manage with smaller set of samples than we we would theoretically need.

Day 2.	Class 1.	Class 2.	Class 3.	CC [bit/s]	FPR [%]	RS [%]
MLP: No prep.	43	22	51	0.41	44	31
MLP: Lap.	47	71	59	0.99	27	19
MLP: Lap.+BP	52	50	66	0.88	23	27
MLP: Lap.+NML	47	44	60	0.76	28	30
MLP: Lap.(AR)	55	82	20	0.80	30	25
MLP: Lap.(Wav)	29	87	43	0.99	34	21
RBF: No prep.	8	31	62	0.19	57	22
RBF: Lap.(F)	3	36	38	0.36	31	63
RBF: Lap.+BP(F)	11	33	24	0.44	18	73
RBF: Lap.+NML(F)	4	31	32	0.40	22	71
RBF: Lap.(AR)	16	64	8	0.64	24	62
RBF: Lap.(Wav)	3	70	19	0.68	23	61

Table 4.7: Collected results for the day 2.

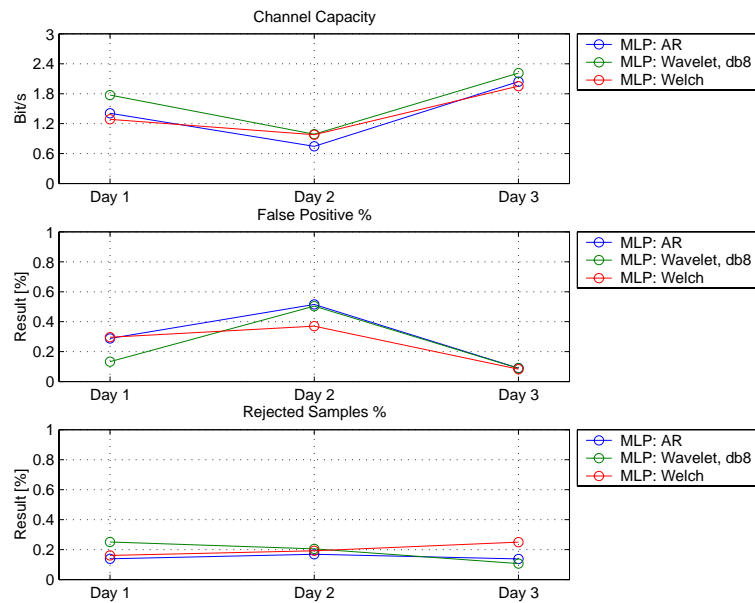


Figure 4.7: Daily performances for the MLP classifier with the different features and  $\rho = 90\%$

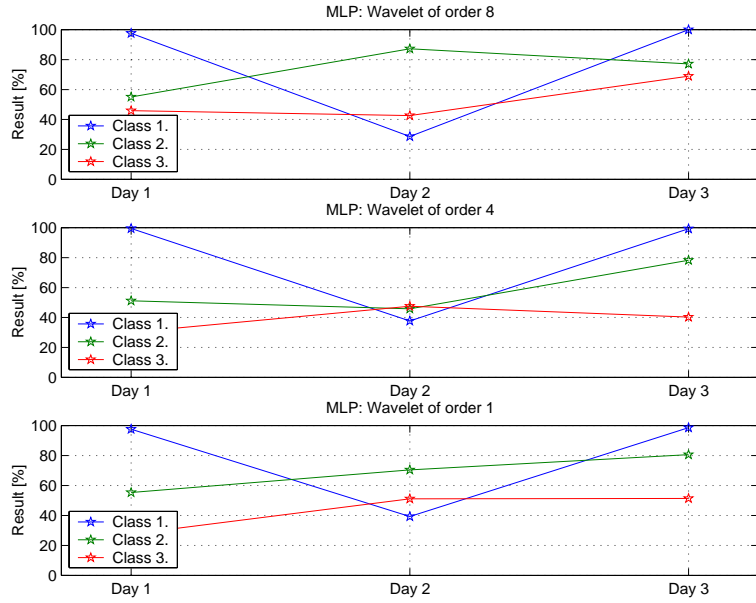


Figure 4.8: Daily results for the wavelet features with  $\rho = 90\%$

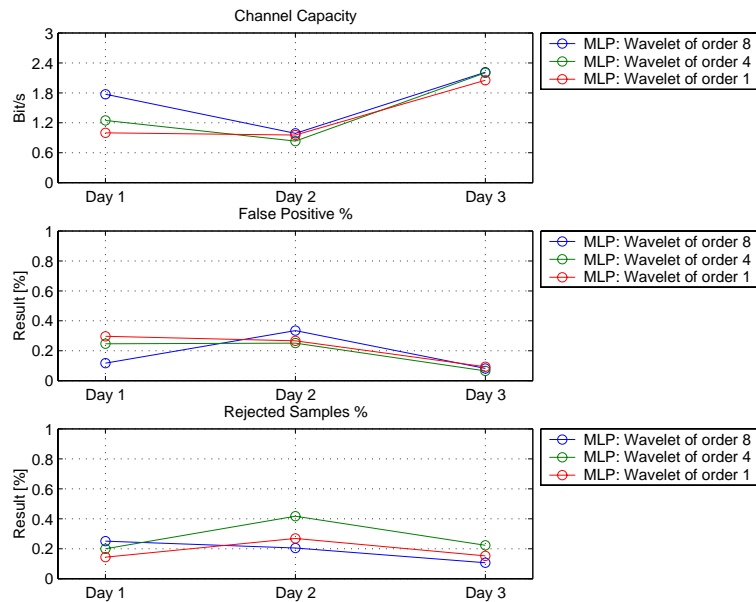


Figure 4.9: Daily results for the wavelet features with  $\rho = 90\%$

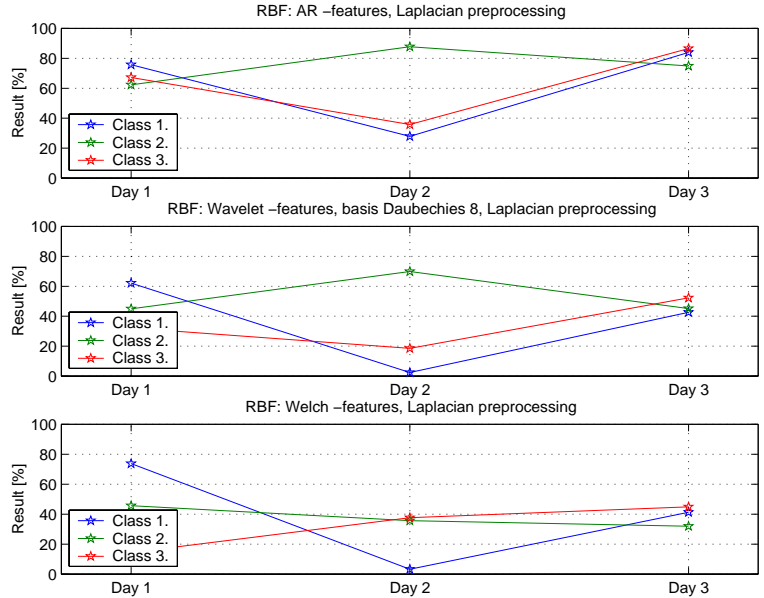


Figure 4.10: Daily results for the RBF classifier with the different features and  $\rho = 50\%$

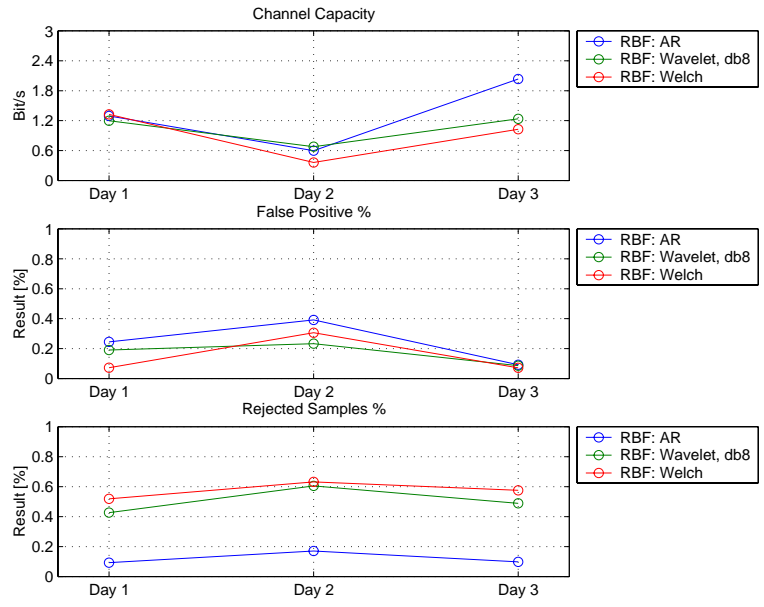


Figure 4.11: Daily performance for the RBF classifier with the different features and  $\rho = 50\%$

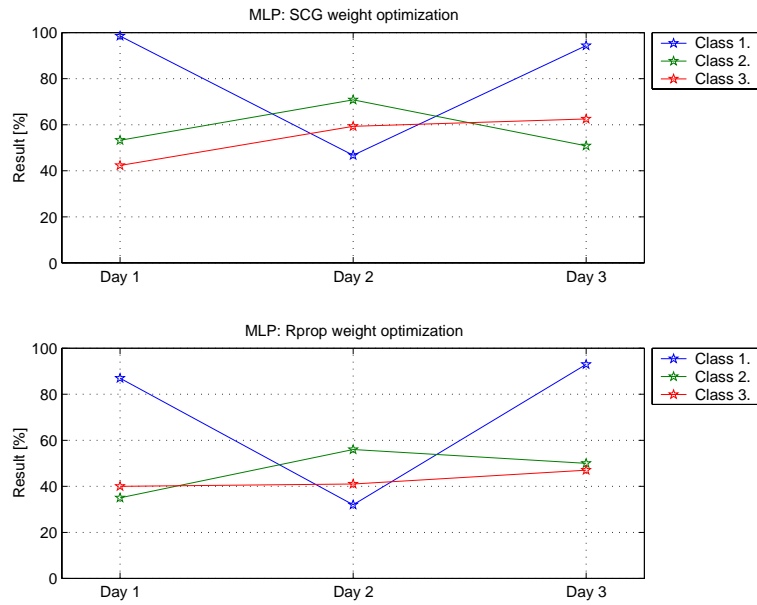


Figure 4.12: Daily results for the weight optimization algorithms

<b>Day 3.</b>	Class 1.	Class 2.	Class 3.	CC [bit/s]	FPR [%]	RS [%]
MLP: No prep.	87	11	20	0.54	37	47
MLP: Lap.	94	51	62	1.95	7	25
MLP: Lap.+BP	95	66	65	1.83	11	15
MLP: Lap.+NML	86	66	50	1.49	15	21
MLP: Lap.(AR)	92	74	67	1.96	11	13
MLP: Lap.(Wav)	100	77	69	2.21	8	11
RBF: No prep.	29	76	18	0.43	45	25
RBF: Lap.(F)	41	32	45	1.03	7	58
RBF: Lap.+BP(F)	24	45	34	0.87	8	63
RBF: Lap.+NML(F)	22	21	50	0.8	11	65
RBF: Lap.(AR)	57	38	54	1.47	3	49
RBF: Lap.(Wav)	43	45	52	1.24	9	49

Table 4.8: Collected results for the day 3.

# Appendix A

## Weight Optimization Algorithms

### A.1 Resilient Propagation (RProp)

Rprop is a local adaptive learning method for optimizing MLP's weights. In this method the influence on weight step is not determined by size of the partial derivatives of the error function but the *sign* of it. The actual weight step is defined by a constant, weight specific *update-value*. The first step in this method is to define the sign of the constant update-value  $\Delta_{ij}$  from the partial derivatives.

$$\Delta w_{ij} = \begin{cases} -\Delta_{ij}, & \text{if } \frac{\partial E}{\partial w_{ij}}(t) > 0 \\ +\Delta_{ij}, & \text{if } \frac{\partial E}{\partial w_{ij}}(t) < 0 \\ 0, & \text{else} \end{cases} \quad (\text{A.1})$$

Then the update-value is further multiplied with a positive or a negative constant, depending on the sign of the product of current and previous partial derivatives

$$\Delta_{ij}^{(t)} = \begin{cases} \eta^+ \cdot \Delta_{ij}^{(t-1)}, & \text{if } \frac{\partial E}{\partial w_{ij}}(t-1) \cdot \frac{\partial E}{\partial w_{ij}}(t) > 0 \\ \eta^- \cdot \Delta_{ij}^{(t-1)}, & \text{if } \frac{\partial E}{\partial w_{ij}}(t-1) \cdot \frac{\partial E}{\partial w_{ij}}(t) < 0 \\ \Delta_{ij}^{(t-1)}, & \text{else} \end{cases} \quad (\text{A.2})$$

where  $0 < \eta^- < 1 < \eta^+$

The values of the increase/decrease factor  $\eta$  are set to 1.2 and 0.5 respectively, based on recommendations from literature, [Rie94].

In the Rprop algorithm is two variables: The first is the starting value of the update-value  $\Delta_0$ , which is usually set to 0.1. The second one is maximum value for the update-value  $\Delta_{max}$ , which purpose is to limit the weights from becoming too large.  $\Delta_{max}$  is usually set to 50.

The benefits of the Rprop is that it is not computationally expensive and the adaptation is not blurred by un-foreseeable gradient behaviour, which means that large values of partial derivated respect to the error function do not 'throw' the search to completely new areas on the error surface.

On the other hand the drawback are that the weight update can be calculated only after sum of all errors is calculated. This can reduce the efficiency on redundant training sets and may cause problems with variable training sets.

## A.2 Scaled Conjugate Gradient (SCG)

SCG is based on the gradient search, where at the each round of iteration the minimum is searched along line given by negative gradient of the error function at the current position. This error minimization proceeds usually very slowly as each new gradient is orthogonal to previous line search direction, as shown in figure (A.3).

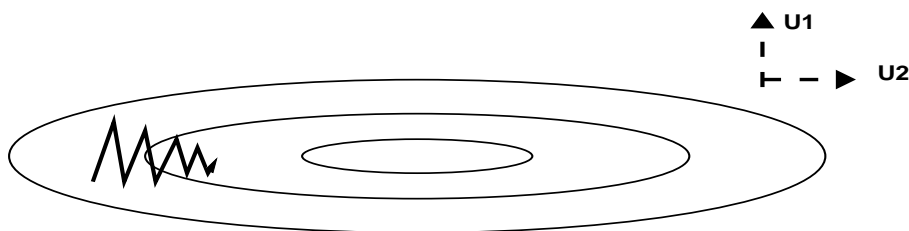


Figure A.1: Problem of the gradient descent: local gradients  $-\nabla E$  do not point towards error minimum, instead they oscillate across “the error valley”

One solution to this problem is to choose new search directions so, that at each step of the iteration the component of the new search direction parallel to the old search direction remains zero (to lowest order) as illustrated in figure (A.2).

This is achieved when equation (A.3) holds.

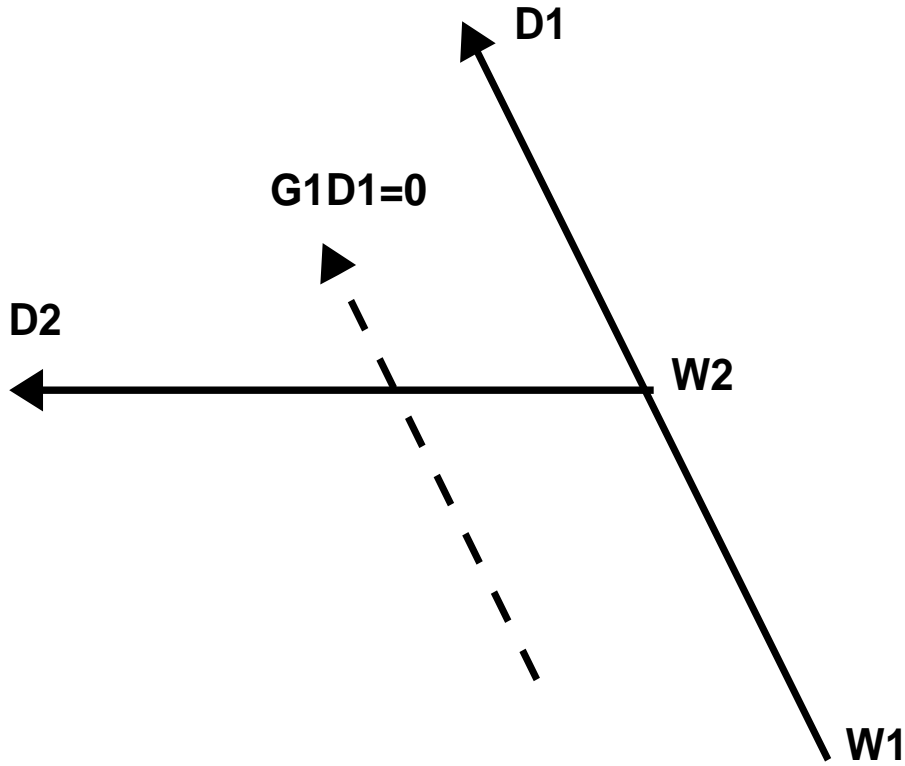


Figure A.2: Conjugate directions:  $w_2$  is minimum point on the line starting at  $w_1$  to direction  $d_1$ . If the component of the gradient of  $w_2$  to direction  $d_2$  parallel to previous line remains zero, then the direction  $d_2$  is said to be *conjugate* to direction  $d_1$

$$\mathbf{d}^{(k+1)T} \mathbf{H} \mathbf{d}^{(k)} = 0, \quad (\text{A.3})$$

where  $\mathbf{d}^{(k+1)T}$  is the new search direction,  $\mathbf{H}$  is the Hessian matrix of the current point in the weight space and  $\mathbf{d}^{(k)}$  is the old search direction. Equation (A.3) holds when  $\mathbf{H}$  is *positive definite*, that is

$$\mathbf{d}^{(k+1)T} \mathbf{H} \mathbf{d}^{(k)} > 0, \quad (\text{A.4})$$

for all  $\mathbf{d}$ .

In the derivation of the *scg* algorithm *quadratic error surface* approximation is used, which gradient is

$$\mathbf{g}(\mathbf{w}) = \mathbf{b} + \mathbf{H}\mathbf{w} \quad (\text{A.5})$$

where  $\mathbf{b}$  and  $\mathbf{H}$  are assumed to be constant and  $\mathbf{H}$  to be positive definite. The actual modification of the weight values is given by equation (A.6).

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \alpha_k \mathbf{d}_k \quad (\text{A.6})$$

where  $\alpha_k$  is step length control parameter,  $\mathbf{w}$  is a point in weight space and  $k$  is current iteration round. Control parameter  $\alpha$  is given by equation (A.7)

$$\alpha_k = \frac{\mathbf{d}_k^T \mathbf{g}(\mathbf{w}_k)}{\mathbf{d}_k^T \mathbf{H} \mathbf{d}_k}. \quad (\text{A.7})$$

It can be shown that if the weight are incremented by using equation (A.6) then the gradient vector  $\mathbf{g}_k$  at  $k$ th step is orthogonal to all previous conjugate directions [Bis95]. That means that after  $W$  steps equal to the dimension of the search space have been taken, we have arrived to the minimum of quadratic error surface.

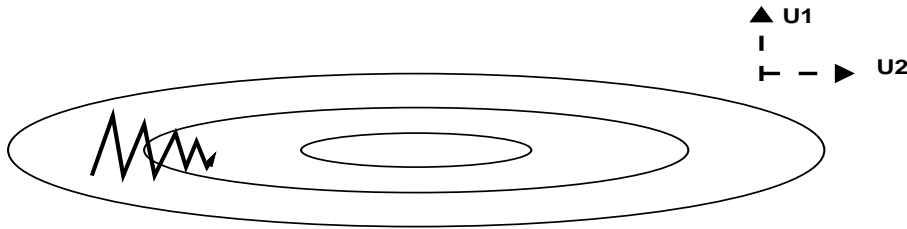


Figure A.3: Problem of the gradient descent: local gradients  $-\nabla E$  do not point towards error minimum, instead they oscillate across “the error valley”

These  $W$  conjugate directions can be found by using equation (A.8)

$$\mathbf{d}_{k+1} = -\mathbf{g}_{k+1} + \beta_k \mathbf{d}_k, \quad (\text{A.8})$$

where coefficients  $\beta$  can be found by using equation (A.9)

$$\beta_k = \frac{\mathbf{g}_{k+1}^T \mathbf{H} \mathbf{d}_k}{\mathbf{d}_k^T \mathbf{H} \mathbf{d}_k}. \quad (\text{A.9})$$

$\mathbf{H}$  is constant only in the quadratic error surfaces, so in general non-quadratic error surfaces the  $\mathbf{H}$  depends on the current weight vector and must be evaluated at each round of iteration. Since this computationally very heavy procedure, it would be a good thing if  $\mathbf{H}$  could be replaced in equations (A.7) and (A.9). Also the line search procedure is computationally heavy. A solution to this is to calculate, instead of the line minimization, the

value  $\mathbf{H}\mathbf{d}_k$  directly by using method called *Fast multiplication by the Hessian* by Møller [Bis95]. In this method  $\mathbf{H}\mathbf{d}_k$  is calculated during normal backpropagation algorithm by first multiplying the input vector by  $\mathbf{d}$  and adding its product with output of the hidden layer to the hidden layer. Problem of this approximation is that in a non-quadratic error function the Hessian may not be positive definite and therefore denominator of equation (A.7) may become negative, which leads to the increase of the error in the iteration step. This problem can be overcome by modifying the Hessian to ensure that it is always positive definite. This is done by adding some multiple of unit matrix to the Hessian. Now equation (A.7) is

$$\alpha_k = \frac{\mathbf{d}_k^T \mathbf{g}(\mathbf{w}_k)}{\mathbf{d}_k^T \mathbf{H}\mathbf{d}_k + \lambda_k \|\mathbf{d}_k\|^2}. \quad (\text{A.10})$$

where  $\lambda_k$  indicates that it varies between the iteration rounds. For the large values of  $\lambda_k$  the step size becomes small. The  $\lambda_k$  for current iteration round is calculated from equation

$$\bar{\lambda}_k = 2 \left( \lambda_k - \frac{\delta_k}{\|\mathbf{d}_j\|^2} \right) \quad (\text{A.11})$$

where the idea is to keep the denominator  $\delta_k$  of equation (A.10) positive by solving the new  $\bar{\lambda}_k$  which keeps the equation (A.10) positive.

### A.3 Forward Selection

Idea of the forward selection is to find a subset of basis functions draw from fixed set of candidates. There is  $2^M - 1$  subsets in a set of size  $M$ , so heuristics must be used to find a small but sufficiently good fraction from the space of all subsets. This algorithm starts from the empty subset to which one subset which most reduces SSE is added at a time until certain criterion is met. In this work the criteria used was *generalized cross-validation (GCV)*. Mathematically this method is expressed as follows. As the basis function is added one at a time, the key equation is the change in the projection matrix  $\mathbf{P}$ . This is given by equation (A.12).

$$\mathbf{P}_{m+1} = \mathbf{P}_m - \frac{\mathbf{P}_m \mathbf{f}_J \mathbf{f}_J^T \mathbf{P}_m}{\mathbf{f}_J^T \mathbf{P}_m \mathbf{f}_J}, \quad (\text{A.12})$$

where  $m$  is number of hidden units,  $J$ th member is added and  $\mathbf{f}_J$  are columns of *design matrix*, which is the matrix of fixed set of candidates. The choice of basis function can be based on finding the greatest decrease in SSE. By combining (A.12) and the equation for the SSE

$$\text{SSE} = \hat{\mathbf{y}}^T \mathbf{P}^2 \hat{\mathbf{y}} \quad (\text{A.13})$$

we get

$$\text{SSE}_m - \text{SSE}_{m+1} = \frac{(\hat{\mathbf{y}}^T \mathbf{P}_m \mathbf{f}_J)^2}{\mathbf{f}_J^T \mathbf{P}_m \mathbf{f}_J} \quad (\text{A.14})$$

The maximum of this difference (over  $1 \leq J \leq M$ ), is used to find the best basis function to add to the network. This algorithm is reasonably fast but it can be speeded up further by using *orthogonal least squares* algorithm. Here is used the fact that every matrix can be factored into the product of a matrix with orthogonal columns and a matrix which is upper triangular. In this method  $\mathbf{f}_J$  is orthogonal to all previous columns and similarly as explained in section (2.4.1) now only  $p$  floating point operations are needed instead of  $p^2$ . Equation (A.13) now becomes

$$\text{SSE}_m - \text{SSE}_{m+1} = \frac{(\hat{\mathbf{y}}^T \tilde{\mathbf{f}}_J)^2}{\tilde{\mathbf{f}}_J^T \tilde{\mathbf{f}}_J}, \quad (\text{A.15})$$

where  $\tilde{\mathbf{f}}_J$  is projection perpendicular to design matrix of already selected basis functions. Performance of this algorithm can be further modestly improved by using *regularisation* which is explained in detail in chapter (2.4.2), though applying regularisation to orthogonal least squares method needs little modification as explained in [Orr96]. Now the difference in SSE is given by

$$\text{SSE}_m - \text{SSE}_{m+1} = \frac{(2\lambda + \tilde{\mathbf{f}}_J^T \tilde{\mathbf{f}}_J)(\hat{\mathbf{y}}^T \tilde{\mathbf{f}}_J)^2}{\lambda + \tilde{\mathbf{f}}_J^T \tilde{\mathbf{f}}_J}, \quad (\text{A.16})$$

where  $\lambda$  is regularisation parameter as explained in section (2.4.2) As explained earlier, basis function are added until GCV criterion starts to rise. Cross-Validation process was explained in section (2.4.2) and by using linear algebra the GCV value is calculated by using equation (A.17)

$$\sigma_{GCV}^2 = \frac{p \hat{\mathbf{y}}^T \mathbf{P}^2 \hat{\mathbf{y}}}{(\text{trace}(\mathbf{P}))^2}, \quad (\text{A.17})$$

where  $p$  is number of training patterns,  $\hat{\mathbf{y}}$  are targets and  $\mathbf{P}$  is projection matrix. From equation (A.17) can be seen that this is just adjusting the average MSE over the training set ( $SSE = \hat{\mathbf{y}}^T \mathbf{P}^2 \hat{\mathbf{y}}$ ). There are several other criteria based on the same principle but in literature [Orr96] this one is probably the most popular.

# Appendix B

## Channel Capacity

### B.1 Channel Capacity Performance Measure

First some basic definitions:

- $\mathbf{A}$  = inputs =  $\{a_i\}$ ,  $i = 1, 2, 3, \dots, r$ . These are the symbols transmitted to the channel.
- $\mathbf{B}$  = outputs =  $\{b_j\}$ ,  $j = 1, 2, 3, \dots, s$ . These are the symbols received from the channel.
- $P(b_j)$ , probability of the output when the input is unknown.
- $P(b_j|a_i)$ , probability of the output when the input is known (the model).
- $P(a_i)$ , *a priori* probability of the input symbols.
- $P(a_i|b_j)$ , *a posteriori* probability of the input symbols by Bayes rule.

$$P(a_i|b_j) = \frac{P(b_j|a_i)P(a_i)}{P(b_j)} \quad (\text{B.1})$$

- Information channel matrix:

$$\begin{pmatrix} P(b_1|a_1) & P(b_2|a_1) & \dots & P(b_s|a_1) \\ P(b_1|a_2) & P(b_2|a_2) & \dots & P(b_s|a_2) \\ \vdots & & & \\ P(b_1|a_r) & P(b_2|a_r) & \dots & P(b_s|a_r) \end{pmatrix} \quad (\text{B.2})$$

The columns correspond to outputs **B** and the rows to inputs **A**. Every row must add up to 1 i.e.  $\sum_{j=1}^s P(b_j|a_i) = 1$ . **This matrix is the same as the confusion matrix of the output of some model!!**

### B.1.1 Definition of channel capacity

Channel capacity of a system is equal to the maximal *mutual entropy* between the inputs and the outputs.  $C = \max_{P(a)} I(\mathbf{A}; \mathbf{B})$ , where

$$\begin{aligned} I(\mathbf{A}; \mathbf{B}) &= H(\mathbf{A}) - H(\mathbf{A}|\mathbf{B}) \\ &= H(\mathbf{B}) - H(\mathbf{B}|\mathbf{A}) \end{aligned} \tag{B.3}$$

Mutual entropy between **A** and **B** means the reduction of uncertainty of **A** when we know **b** and vice versa. In the equation (B.3)

- $H(\mathbf{A})$  = average information (or uncertainty) of input alphabet *before* the output is known.
- $H(\mathbf{B})$  = average information (or uncertainty) of output alphabet *before* the input is known.
- $H(\mathbf{A}|\mathbf{B})$  = average information (or uncertainty) of input alphabet *after* the output is known.
- $H(\mathbf{B}|\mathbf{A})$  = average information (or uncertainty) of output alphabet *after* the input is known.
- $H(\mathbf{A}) - H(\mathbf{A}|\mathbf{B}) = H(\mathbf{B}) - H(\mathbf{B}|\mathbf{A})$  average information (or uncertainty) of input alphabet *provided* by the channel.

The relationships of the entropies in a classifier type of system are shown in figure (B.1).

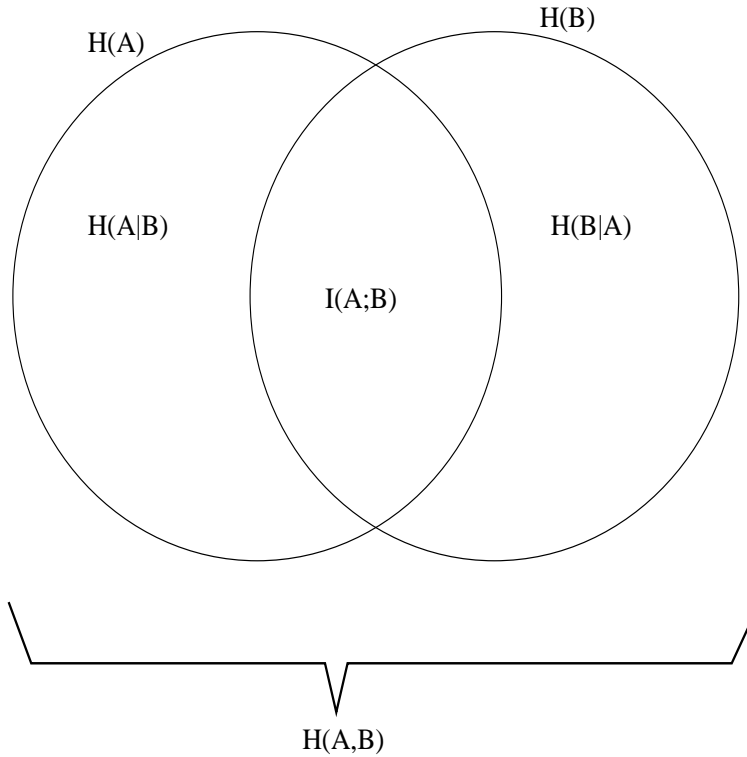


Figure B.1: The components of the total entropy of the system

### B.1.2 Channel Capacity for ABI

In ABI some classification are rejected if the probability do not exceed pre-defined threshold. Therefore we have confusion matrices like this (taken from [dRM00], the latter one).

$$\begin{pmatrix} 0.92 & 0.0 & 0.0 \\ 0.0 & 0.51 & 0.05 \\ 0.0 & 0.09 & 0.56 \end{pmatrix} \quad (\text{B.4})$$

This matrix is not uniform, which makes the computation more difficult. The answer here is to use *erasure channel* -approach ([Slo92]), where some of the input will be lost to the channel with a probability. Erasure channel is applied by adding additional column to the channel information matrix (= confusion matrix), which corresponds to the lost percentage. Let's transform the matrix (B.4).

$$\begin{pmatrix} 0.92 & 0.0 & 0.0 & 0.08 \\ 0.0 & 0.51 & 0.05 & 0.44 \\ 0.0 & 0.09 & 0.56 & 0.35 \end{pmatrix} \quad (\text{B.5})$$

Now the channel capacity as the function of  $P(a_i)$  is

$$H(\mathbf{B}) - H(\mathbf{B}|\mathbf{A}) \quad (\text{B.6})$$

$$= \sum_j^s P(b_j) \log P(b_j) - \sum_i^r P(a_i) \sum_j^s P(b|a_i) \log \frac{1}{P(b|a_i)} \quad (\text{B.7})$$

$$= \sum_j^s \left( \sum_i^r P(b_j|a_i) P(a_i) \log P(b_j|a_i) P(a_i) \right) - \sum_i^r P(a_i) \sum_j^s P(b|a_i) \log \frac{1}{P(b|a_i)} \quad (\text{B.8})$$

The channel capacity was defined as maximum mutual information respect to the input alphabets  $P(a_i)$  so now the equation (B.8) is maximized respect to the  $P(a_i)$  by some optimization method. In this work the optimization was done by MatLab software.

# Bibliography

- [Ber29] H. Berger. On the electroencephalograph of the man. *Arch. Psychiatrie. Nervenkrankheiten*, 1929.
- [Bir00] Niels Birbaumer. The thought translation device (tt) for completely paralyzed patients. *IEEE Transactions on Rehabilitation Engineering*, 8(2):190–193, 2000.
- [Bis95] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.
- [Ci193] P.J. Cilliers. A vep-based computer interface for c2-quadruplegics. In *IEEE Conference on Electronic Devices for the Disabled – Beyond 2000*, 1993.
- [dRM00] Jose del R. Millan. Local neural classifier for eeg-based recognition of mental tasks. In *IEEE-INNS-ENNS International Joint Conference on Neural Networks*, 2000.
- [F.96] Babiloni F. Spline laplacian estimate of eeg potentials over magneto-resonance constructed scalp surface model. *Electroencephalography and Clinical Neurophysiology*, 98, 1996.

- [Fre75] W. J. Freeman. *Mass Action in the Nervous System*. Academic Press, 1975.
- [Hir90] A. H. Hiraiwa. Eeg topography recognition by neural networks. *IEEE Eng. Med. Bio. Mag.*, 9(3):39–42, September 1990.
- [HO99] Aapo Hyvärinen and Erkki Oja. Independent component analysis: A tutorial. Technical report, Helsinki University of Technology, Laboratory of Computer and Information Science, 1999.
- [Hug95] Kenneth Hugdahl. *Psychophysiology, The Mind-Body Perspective*. Harvard University Press, 1995.
- [IEC93] Barrie W. Ifeachor Emmanuel C., Jervis. *Digital Signal Processing: A Practical Approach*. Addison-Wesley, 1993.
- [IEE00] IEEE. *IEEE Transactions on Rehabilitation Engineering*, volume 8. IEEE, June 2000.
- [Ilm00] Risto Ilmoniemi. *The Structure and Operation of the Human Brain*, 2000.
- [Kei90] Z.A. Keirn. Man-machine communications through brain-wave processing. *IEEE Engineering in Medicine and Biology Magazine*, March 1990.
- [Kuh8Aa] W.N. Kuhlman. Eeg feedback training: Enhancement of somatosensory cortical activity. *Electroencephalography and Clinical Neurophysiology*, 45:290–294, 1978A.
- [Kuh8Ab] W.N. Kuhlman. Functional topography of the human mu rhythm. *Electroencephalography and Clinical Neurophysiology*, 44:83–93, 1978A.

- [LE88] Farwell L.A. and Donchin E. Talking off the top of your head: towards a mental prosthesis utilizing event-related brain potentials. *Electroencephalography and Clinical Neurophysiology*, 70:510–523, 1988.
- [Mal98] Stephane Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 1998.
- [McF93] D.J. McFarland. An eeg-based method for graded cursor control. *Psychobiology*, 21(1):77–81, 1993.
- [McF98] D.J. McFarland. Eeg-based communication and control: Short-term role of feedback. *IEEE Transactions on Rehabilitation Engineering*, 6(1):7–11, 1998.
- [Nie99a] Ernst Niedermayer, editor. *Electroencephalography*, chapter 4, pages 77–91. Lippincott Williams & Wilkins, 1999.
- [Nie99b] Ernst Niedermayer, editor. *Electroencephalography*, chapter 53, pages 958–968. Lippincott Williams & Wilkins, 1999.
- [Nie99c] Ernst Niedermayer, editor. *Electroencephalography*, chapter 32, pages 637–659. Lippincott Williams & Wilkins, 1999.
- [Nie99d] Ernst Niedermayer, editor. *Electroencephalography*, chapter 7, page 123. Lippincott Williams & Wilkins, 1999.
- [Nie99e] Ernst Niedermayer, editor. *Electroencephalography*. Lippincott Williams & Wilkins, 1999.
- [O’H98] Donald E. O’Hair. Biofeedback: Review, history and application. Technical report, Discrete Data Systems, Inc, 1998.

- [Orr96] Mark J. L. Orr. Introduction to radial basis function networks. Technical report, Centre for Cognitive Science, University of Edinburgh, 1996.
- [Pfu93] G. Pfurtscheller. Brain-computer interface – a new communication device for handicapped persons. *Journal of Microcomputer Applications*, 16:293–299, 1993.
- [Pfu96] G. Pfurtscheller. Automated feature selection with a distinction sensitive learning vector quantizer. *Neurocomp.*, 11:19–29, 1996.
- [Pfu00] G. Pfurtscheller. Current trends in graz brain-computer interface (bci) research. *IEEE Transactions on Rehabilitation Engineering*, 8(2):216–219, 2000.
- [PR98] William D. Penny and Stephen J. Roberts. Imagined hand movements identified from the eeg mu-rhythm. Technical report, Imperial College of Science Technology & Medicine, London, UK, 1998.
- [PRS99] William D. Penny, Stephen J. Roberts, and Maria J. Stokes. Eeg-based communication: a pattern recognition approach. Technical report, Imperial College of Science Technology & Medicine, London, UK, 1999.
- [Rie94] Martin Riedmiller. Advanced supervised learning in multi-layer perceptrons. Technical report, Institut für Logik, Komplexität und Deduktionssysteme, University of Karlsruhe, 1994.
- [Ris99] Jorma Rissanen. Stochastic complexity in statistical inquiry. Technical report, Helsinki University of Technology, 1999.

- [RP99] Stephen J. Roberts and William D. Penny. Real-time brain-computer interfacing: a preliminary study using bayesian learning. Technical report, Imperial College of Science Technology & Medicine, London, UK, 1999.
- [Slo92] A. Sloane, editor. *Claude Elwood Shannon, Collected Papers*. IEEE Press, 1992.
- [Sut92] E.E. Sutter. The brain response interface: communication through visually-induced electrical brain responses. *Journal of Microcomputer Applications*, 15:31–45, 1992.
- [The92] Charles W. Therrien. *Discrete Random Signals and Statistical Signal Processing*. Prentice-Hall, 1992.
- [Wol91] J.R Wolpaw. An eeg-based brain-computer interface for cursor control. *Electroencephalography and Clinical Neurophysiology*, 78:252–259, 1991.
- [Wol94] J.R Wolpaw. Multichannel eeg-based brain-computer communication. *Electroencephalography and Clinical Neurophysiology*, 90:444–449, 1994.
- [Wol00] Jonathan R. Wolpaw. Brain-computer interface research at the wadsworth center. *IEEE Transactions on Rehabilitation Engineering*, 8(2):222–226, 2000.
- [WV00] Jonathan R. Wolpaw and Theresa M. Vaughan. Brain-computer interface technology: A review of the first international meeting. *IEEE Transactions on Rehabilitation Engineering*, 8(2):164–172, 2000.
- [Zha98] Z. Zhang. Comparison between geometry-based and gabor-wavlet-based fa-

cial expression recognition using multi-layer perceptrons. In *Third International Conference on Automatic Face & Gesture Recognition (FG'98)*, 1998.