

**A COMPARISON OF FEATURE EXTRACTION AND
CLASSIFICATION METHODS FOR MOTOR IMAGERY EEG
SIGNALS**

By

Cristian H. Alzate

A thesis submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in

ELECTRICAL ENGINEERING

UNIVERSITY OF PUERTO RICO
MAYAGÜEZ CAMPUS

December, 2014

Approved by:

Vidya Manian, Ph.D.
Chairman, Graduate Committee

Date

Domingo Rodriguez, Ph.D.
Member, Graduate Committee

Date

Shawn D. Hunt, Ph.D.
Member, Graduate Committee

Date

Pedro Vasquez, Ph.D.
Graduate Studies Representative

Date

Pedro I. Rivera-Vega, Ph.D.
Department Chairperson

Date

Abstract of Thesis Presented to the Graduate School
of the University of Puerto Rico in Partial Fulfillment of the
Requirements for the Degree of Master of Science

**A COMPARISON OF FEATURE EXTRACTION AND
CLASSIFICATION METHODS FOR MOTOR IMAGERY EEG
SIGNALS**

By

Cristian H. Alzate

December 2014

Chair: Dr. Vidya Manian

Department: Electrical and Computer Engineering Department

Recent development in the field of biomedical science has resulted in a proliferation of Brain Computer Interfaces (BCI), signals are acquired and processed more efficiently. Using the Event-related de-synchronization (ERD) concept, where the μ frequency appears in the motor and sensory cortex when a subject is doing or even imagining a movement, a new method to extract features from this type of EEG signals called Common Space Analytic Pattern (CSAP) is used. CSAP is based on a spatial filter which recovers underlying source signals near the motor cortex which are indicative of motor imagery and with the hidden Markov model (HMM) approach as classifier, outstanding results are achieved for binary Motor Imagery EEG signals. In this work, Support Vector Machine is used with CSAP to classify binary motor imagery EEG signals, its performance is compared with Hidden Markov Model. Both classifiers are used in a BCI called path speller that executes in real time. Experiment shows that SVM has a very good accuracy, but HMM is more accurate, faster and reaches higher confidence level (high probability of a trial to belong to a known class). This also was the case with the path speller BCI, where HMM with CSAP was more efficient than SVM.

Resumen de tesis presentado a la Escuela Graduada
de la Universidad de Puerto Rico como requisito parcial de los
requerimientos para el grado de Maestría en Ciencias

**UNA COMPARACIÓN DE MÉTODOS DE EXTRACCIÓN DE
CARACTERÍSTICAS Y CLASIFICACIÓN DE SEÑALES EEG DE
IMAGINACIÓN MOTRIZ**

Por

Cristian H. Alzate

Diciembre 2014

Consejero: Dr. Vidya Manian

Departamento: Ingeniería Eléctrica y Computadoras

Los recientes avances en el campo de la biomedicina ha dado lugar a una proliferación de Interfaces cerebro-computador (BCI), las señales son adquiridas y procesadas mas eficientemente. Usando el concepto de la desincronización basada en eventos (ERD), donde la frecuencia μ aparece en la corteza motora y sensorial cuando una persona está haciendo o incluso imaginando un movimiento, un nuevo método para extraer características de este tipo de señales EEG llamado patrón analítico de espacio común (CSAP) ha sido usado. CSAP se basa en un filtro espacial el cual recupera las señales de origen subyacentes cerca de la corteza motora, que son indicativos de la imaginación de movimientos y con el enfoque del modelo oculto de Markov (HMM) como clasificador, excelentes resultados se lograron para las señales EEG de imaginación de movimientos binarias. En este trabajo, el SVM se utiliza con el CSAP para clasificar dicho tipo de señales, su desempeño es comparado con los modelos ocultos de Markov. Ambos clasificadores se utilizan en un BCI llamado “path speller” que se ejecuta en tiempo real. El experimento demuestra que SVM tiene una muy buena precisión, pero HMM es más preciso, más rápido y alcanza un mayor nivel de confianza (alta probabilidad de un intento de pertenecer a una clase

conocida). Este también fue el caso del BCI, donde CSAP con HMM fue más eficaz que SVM.

Copyright © 2014

by

Cristian H. Alzate

This work is dedicated to my mother and all those who believe in me

Acknowledgements

I would like to thank Dra Vydya Manian to make possible this work. I would also like to thank Dr. Todd Coleman and Justin Tantiogloc for their constant support and advices, thanks also to Mr Samuel Rosario and Victor Asencio for their technical support and the Laboratory for Applied Remote Sensing and Image Processing, University of Puerto Rico for providing the software platforms for data processing.

Table of Contents

Abstract in English	ii
Abstract in Spanish	iii
Copyright	v
Dedication	vi
Acknowledgements	vii
List of Tables	xi
List of Figures	xii
List of Abbreviations	xiv
List of Symbols	xv
1 INTRODUCTION	1
1.1 Motivation	1
1.2 Outline	2
1.3 Objectives	2
1.3.1 General Objective	2
1.3.2 Specific Objectives	2
2 THEORETICAL BACKGROUND	3
2.1 EEG Signals	3
2.2 Empirical Mode Decomposition (EMD)	6
2.3 Principal Component Analysis (PCA)	9
2.4 Common Spatial Pattern (CSP)	9
2.5 Support Vector Machine (SVM)	12
2.6 Naive-Bayes	14
2.7 Neural Networks	16

2.8	Hidden Markov Model	17
3	METHODOLOGY	19
3.1	Signal Acquisition	20
3.1.1	Hardware	20
3.1.2	Software	22
3.2	Signal Preprocessing	23
3.3	Signal Feature Extraction	25
3.4	Classification	29
3.4.1	Hidden Markov Model	29
3.4.2	Support Vector Machine	30
3.5	Implementation	31
3.5.1	Ball display	32
3.5.2	Brain-Computer Interface: Pathspeller	33
3.6	Experiment	35
4	EXPERIMENTAL RESULTS	36
4.1	Emotiv's EPOC Neuroheadset Experiments	36
4.1.1	EEG Classification using SOM	38
4.1.2	EEG Classification using SVM	39
4.2	BrainAmp By Brain Products GmbH experiments	44
4.2.1	Data acquisition and Processing	44
4.2.2	Experimental Results	47
5	PATH SPELLER RESULTS	53
5.1	Accuracy	53
5.2	Processing time	53
5.3	Path Speller	55
6	CONCLUSIONS	58
6.1	Contributions	58

6.2 Future Work	58
Bibliography	60

List of Tables

3.1	BrainAmp's Technical Specifications	23
4.1	SOM Classification Results for experienced subjects	39
4.2	SOM Classification Results for beginner subjects	40
4.3	SVM Classification Results for experienced subjects	42
4.4	SVM Classification Results for beginner subjects	43
4.5	Results using CSP-SVM classifier	49
4.6	Results using Naive-Bayes classifier	51
4.7	Results for all the classifiers	51
5.1	Results for Subject A	54
5.2	Results for Subject B	54
5.3	Classification time in milliseconds	55
5.4	Computer characteristic	55

List of Figures

2.1	Example of EEG Signal.	5
2.2	Example of EMD algorithm.	8
2.3	CSP Example.	11
2.4	SVM Mapping Example.	14
2.5	Model of a neuron.	16
3.1	Methodology Flowchart	20
3.2	Brain Map and channel selection.	21
3.3	BrainAmp and Powerpack	22
3.4	Kit to reduce electrodes impedance	24
3.5	Methodology Diagram.	25
3.6	EEGClassifier Schematic diagram.	32
3.7	Visual stimuli used in Training phase	33
3.8	BCI Model.	33
3.9	Pathspeller BCI.	34
4.1	Emotiv EPOC Neuroheadset electrode placement illustration	36
4.2	Emotiv Cognitive Suite Cube	37
4.3	Clustering using PCA	41
4.4	Spectrum and Spectrogram of C3 Electrode using Motor Imagery for Right Hand.	45
4.5	CSP applied to original and cleaned EEG data.	47
4.6	NeuroLab Graphical User Interface	48
4.7	ROC Curves for SVM Classifier.	50
4.8	ROC Curves for Naive-Bayes Classifier.	52

4.9	ROC Curves comparison for both Classifiers.	52
5.1	Results of HMM classifier in the Pathspeller BCI.	56
5.2	Results of SVM classifier in the Pathspeller BCI.	57

List of Abbreviations

EMD	Empirical Mode Decomposition.
MEMD	Multivariate Empirical Mode Decomposition.
SOM	Self-Organizing Map.
EEG	Electroencephalography.
SVM	Support Vector Machine.
RBF	Radial Basis Function.
ADD	Attention Deficit Disorder.
IMF	Intrinsic Mode Function.
ROC	Receiver Operating Characteristic.
RBF	Radial Basis Function.
PCA	Principal Component Analysis.
SVD	Singular Value Decomposition.
ERD	Event-related de-synchronization.
CSSP	Common Spatio-Spectral Pattern.
CSSSP	Common Sparse Spectral Spatial Pattern.
GUI	Graphical User Interface.
FFT	Fast Fourier Transform.
STFT	Short-Time Fourier Transform.
CSAP	Common Space Analytic Pattern
HMM	Hidden Markov Model.
BCI	Brain Computer Interface.
FFG	Forney factor graph.

List of Symbols

t	Time (seconds).
ms	Time (milliseconds).
Hz	Frequency (Hertz).
KHz	Frequency (Kilohertz).
V	Voltage (Volts).
μV	Voltage (Microvolts).

Chapter 1

INTRODUCTION

1.1 Motivation

Brain signals are stochastic and complex non-stationary, therefore, extracting features effectively by some simple analysis methods in time or frequency domain becomes difficult. It is very necessary to have new tools that make the extraction of information from these signals easier, accurate and faster. For example, using EEG signals, with an optimum extraction of information allow new human-computer interfaces to be developed for brain computer interfaces. This information can be very useful to determine how the user is reacting in real-time. This can help people with major physical disabilities by helping them with their basic activities such as communicating, by providing functional interfaces for non-verbal communication when normal communication is defective.

EEG is a method to capture brain signals with an excellent temporal resolution, non-invasiveness, usability and low cost. Using the ERD reflected in μ frequency which appears in the motor and sensory cortex when a subject is only imagining a movement, new methods to classify this signals were proposed with an outstanding accuracy and speed. In this work, the SVM classifier using the Common Analytic Space Patter approach proposed by [1] is used to classify motor imagery signals in real time, its accuracy, timing and computational cost is compared with a classifier based on Hidden Markov Model proposed by the same author.

1.2 Outline

In order to understand many of the concepts dealt with in the present document a brief description of the methods and algorithms is given in Chapter 2. The stages required for the satisfactory completion of the objectives are described in Chapter 3. A study of some characteristics of the EEG signals, different feature extraction algorithms and classifiers applied to EEG signals are given in Chapter 4. The efficiency of SVM and HMM classifiers for real time application, the comparison, the computational cost as well its accuracy are given in Chapter 5. Finally, the conclusions and the direction for further development and improvement is given in Chapter 6.

1.3 Objectives

1.3.1 General Objective

- Implement efficient feature extraction and classification methods for binary Motor Imagery EEG signals.

1.3.2 Specific Objectives

- Compare the performance of the feature extraction and classification methods in real-time.
- Implement the BCI with CSAP and SVM Classifier.
- Compare the performance of the BCI path speller using SVM classifier with BCI using HMM Classifier.

Chapter 2

THEORETICAL BACKGROUND

In this chapter, the fundamental concepts, the signal types and the algorithms description are presented. The algorithms of PCA, EMD and CSP are used for extract information. HMM, SVM and Neural Networks algorithms are used in classification stage.

2.1 EEG Signals

Electroencephalography or EEG, are signals representing the brain's electrical activity acquired by placing electrodes on a subject's scalp. This electrical signals are the result of the synaptic activity of millions of neurons summed, or nerve cells. One example of this signal is show in figure 2.1 . EEG spectral content is divided into five major frequency bands, called brain rhythms [2]. These brain waves are:

- Delta (δ) Waves: 0.5–3.5 Hz. Slowest and highest in amplitude of the brain waves, delta waves are associated with deep sleep and unconsciousness and are dominant in newborns and infants up to one year old [3]. Among many things, deep sleep is important for the healing process, as its linked with deep healing and regeneration.
- Theta (θ) Waves: 3.5–7.5 Hz. Theta waves appear dominant in light sleep, deep relaxation and people with Attention Deficit Disorder, or ADD. It is the

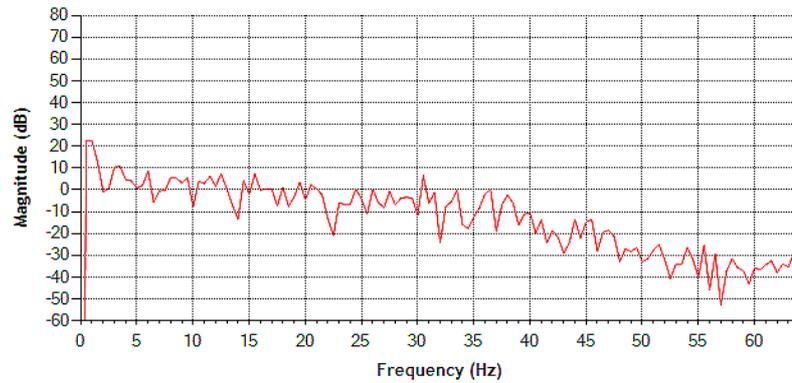
realm of the subconsciousness and only experienced momentarily as the subject drifts off to sleep from Alpha and wake from deep sleep (from Delta).

- Alpha (α) Waves: 7.5–12.9 Hz. Appear in normal adults during wakefulness and mental inactiveness [3]. They are best seen with eyes closed and most pronounced in occipital locations. It is the gateway to the subconscious mind and lies at the base of your conscious awareness. The voice of Alpha is the intuition, which becomes clearer and more profound the closer it gets to 7.5 Hz.
- Beta (β) Waves: 13–30 Hz. Characteristic in normal adults during mental activity. Since this type of wave corresponds to mental engagement, they are best seen in the frontal lobe, responsible for higher mental function [3].
- Gamma (γ) Waves: >30 Hz. This range is the most recently discovered and is the fastest frequency at above 30 Hz. While little is known about this state of mind, initial research shows Gamma waves are associated with bursts of insight and high-level information processing [3].

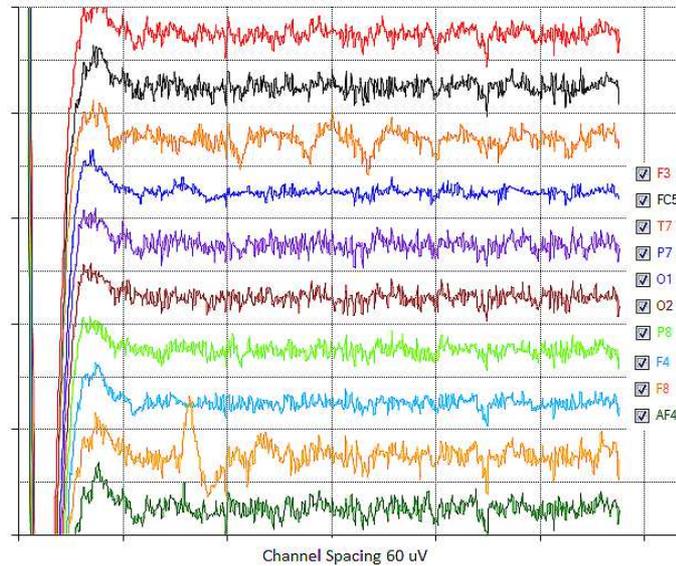
The EEG Signals could be formalized as:

$$\{E_n\}_{n=1}^N \in \mathfrak{R}^{ch \times time} \quad (2.1)$$

Where N is the number of trials, ch the number of channels and time is the range of time domain. For electroencephalography based BCI, motor imagery is considered as one of the most effective ways, Motor Imagery is a mental process of a motor action. It includes preparation for movement, passive observations of action and mental operations of motor representations implicitly or explicitly. The ability of an individual to control his EEG through imaginary mental tasks enables him to control



(a) In Frequency



(b) In Time

Figure 2.1 : Example of EEG Signal.

devices through a brain machine interface (BMI) or a brain computer interface (BCI). In the imagination of limb movement, suppression of EEG signals happens in the specific region of the motor and somatosensory cortex due to loss of synchrony in μ and β bands, classically defined in the 12-16 Hz and 18-24 Hz respectively, is termed event-related de-synchronization (ERD) [4].

The BMI/BCI is a direct communication pathway between the brain and an external

device. BMI/BCI are often aimed at assisting, augmenting, or repairing human cognitive or sensory-motor functions. The methodologies of BMI/BCI can be separated into two approaches:

1. Invasive (or partially-invasive) BMI/BCI: Electrocorticography (ECoG).
2. Non-invasive BMI/BCI: EEG, MEG, MRI fMRI, etc.

Invasive and partially-invasive BCIs are accurate. However there are risks of the infection and damage. Furthermore, it requires the operation to set the electrodes in the head. On the other hand, non-invasive BCIs are inferior than invasive BCIs in accuracy, but costs and risks are very low.

2.2 Empirical Mode Decomposition (EMD)

Empirical Mode Decomposition (EMD) is an algorithm designed for multiscale decomposition and time-frequency analysis of Real-World Signals [5], whereby the original signal is modeled as a linear combination of intrinsic oscillatory modes, called intrinsic mode functions (IMFs), The IMFs are defined so as to exhibit locality in time and to represent a single oscillatory mode; In other words, the EMD algorithm decomposes the original signal into a finite set of amplitude and/or frequency-modulated components, termed IMFs. According to [6], a real signal can be expressed by :

$$x(t) = \sum_{i=1}^N c_i(t) + r(t). \quad (2.2)$$

where $r(t)$ is the monotonic residue signal and $\{c_i(t)\}_{i=1}^N$ the IMFs, having symmetric upper and lower envelopes, the number of zero crossing and vertices differing at most by one defines IMFs $c_i(t)$; ensuring well-behaved intrinsic oscillations. The sifting algorithm is an iterative process to extract the IMFs [6]. The standard EMD algorithm is:

Algorithm 1. The standard EMD Algorithm

1. Find the locations of all the extrema of $x(t)$.
2. Interpolate between all the minima (respectively maxima) to obtain the lower signal envelope, $e_{min}(t)$ (respectively $e_{max}(t)$).
3. Compute the local mean $m(t) = [e_{min}(t) + e_{max}(t)]/2$.
4. Subtract the mean from the signal to obtain the ‘Oscillatory mode’ $s(t) = x(t) - m(t)$.
5. If $s(t)$ obeys the stopping criteria, then $d(t)$ is defined as $s(t)$ as an IMF, otherwise set $x(t) = s(t)$ and repeat the process from step 1.

Once the first IMF is acquired, the remaining IMFs can be extracted by applying the same process iteratively to the outstanding $r(t) = x(t) - d(t)$. When the above condition for an IMF is encountered for S successive times, that is when the standard stopping criterion terminates the sifting.

In Figure 2.2, an example of the EMD algorithm is shown. In figure 2.2 (a) a cosine signal is shown with amplitude 1 and frequency 60 Hz, in fig. 2.2 (b) the same signal with amplitude 2 and frequency 120 Hz. Then, to test the EMD algorithm, signals A and B were added in fig. 2.2 (c). When the algorithm is applied, the signal of fig. 2.2 (c) was divided into 2 IMFs (figure 2.2 (d) and 2.2 (e)) and a Residue (figure 2.2 (f)).

When the signal is multidimensional, the local minima and maxima may not be defined directly because the fields of complex numbers and quaternions are not ordered [7]. To deal with these problems, the multiple real-valued projections of the signals was proposed in [8].

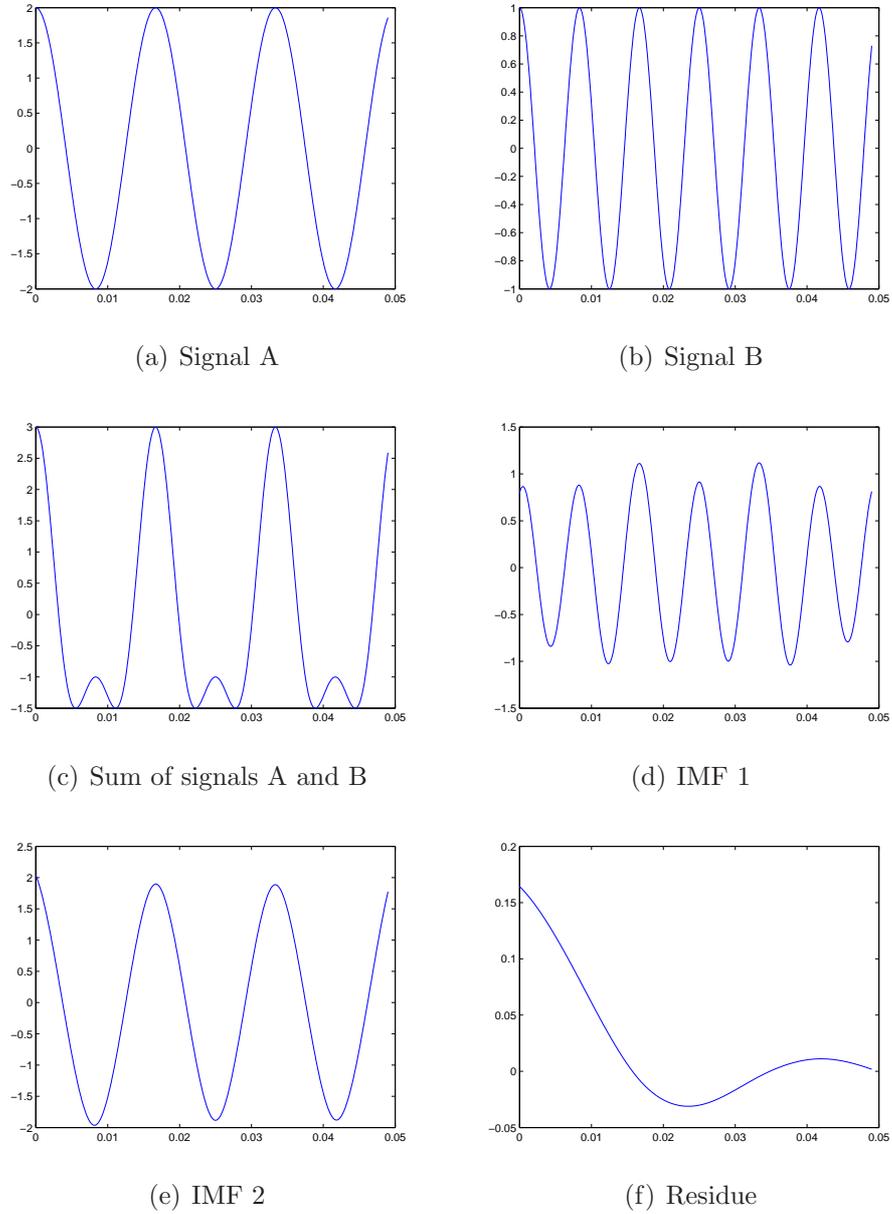


Figure 2.2 : Example of EMD algorithm.

In EEG signals, EMD is very useful for removing some of the IMFs, the reconstructed signal can carry more useful information than the original. The first complex extension of EMD was proposed in [9]. An extension of EMD to analyze complex/bivariate data which operates fully in the complex domain was first proposed in [10], termed rotation invariant EMD (RI-EMD). An other processing method which gives more

accurate values of the local mean is the bivariate EMD (BEMD) [11], where the envelopes corresponding to multiple directions in the complex plane are generated, and then averaged to obtain the local mean.

2.3 Principal Component Analysis (PCA)

One of the most successful techniques that have been used in signal recognition and compression is the Principal Component Analysis, known as PCA. It is a statistical process also known as factor analysis. The need to describe data efficiently gives direction to PCA, which is to minimize the large dimensionality of the data space to smaller intrinsic dimensionality of feature space. This is the case when there is a strong correlation between observed variables [12].

The goals of PCA are [13]:

- Extract the most important information from the data set.
- Compress the size of the data set by keeping only the important information.
- Simplify the description of the data set.
- Analyze the structure of the observations and the variables.

The first principal component is required to have the largest possible variance, the second, is computed under the constraint of being orthogonal to the first component and have the largest possible inertia. The other components are computed likewise [13].

2.4 Common Spatial Pattern (CSP)

The CSP is a spatial filter method, widely used in neuroscience as a linear transformation to project the multi-channel EEG data into low-dimensional spatial subspace with a projection matrix; each row consists of weights for channels. This transformation can maximize the variance of two-class signal matrices based on the

simultaneous diagonalization of the covariance matrices of both classes (in this case, Left and Right). The Common Spatial Patterns algorithm was firstly suggested for classification of multi-channel EEG during imagined hand movements in [14]. The CSP method has shown its efficacy in extracting topographic pattern of brain rhythm modulations, also known as ERD. This spatial filter, according to [15], must only be applied to the informative frequency bands (μ and β bands), which is specific to each subject. In [16] the CSP has been extended to multi-class problems.

The details of the algorithm are described as follows [17] and [18]. Lets say X_l and X_r are two preprocessed EEG matrices under two conditions with dimensions $N \times T$, where N is the number of channels and T is the number of samples in each channel. The normalized covariance matrix of the EEG can be represented as:

$$C_l = \frac{X_l X_l^T}{\text{trace}(X_l X_l^T)} \quad C_r = \frac{X_r X_r^T}{\text{trace}(X_r X_r^T)} \quad (2.3)$$

X^T is the transpose matrix of X and the function trace computes the sum of the diagonal elements in the given input matrix. By averaging over all the trials of each group, the averaged normalized covariance \overline{X}_l and \overline{X}_r are calculated

$$C = \overline{X}_l + \overline{X}_r \quad (2.4)$$

The composite covariance matrix and its eigenvalue decomposition is given by

$$C = F_0 \Sigma F_0^T \quad (2.5)$$

Where F_0 is the matrix of eigenvectors and Σ is the diagonal matrix of eigenvalues.

The whitening transformation

$$P = \Sigma^{-\frac{1}{2}} F_0^T \quad (2.6)$$

Equalizes the variances in the space spanned by the eigenvectors in F_0 . With the simultaneous diagonalization of whitened covariance matrices the CSP can be extracted

$$S_l = P\overline{X}_lP^T \quad S_r = P\overline{X}_rP^T \quad (2.7)$$

S_l and S_r have common eigenvectors and the sum of corresponding eigenvalues for the two matrices will always be one, then this resulting decomposition maximizes the differentiation between two groups of data.

$$S_l = U\lambda U^T \quad S_r = U(1 - \lambda)U^T \quad (2.8)$$

The CSP projection matrix will then be $W_{csp} = (U^T P)$. An graphic example is shown in Fig. 2.3 , where class 1 points are shown in blue and class 2 in red.

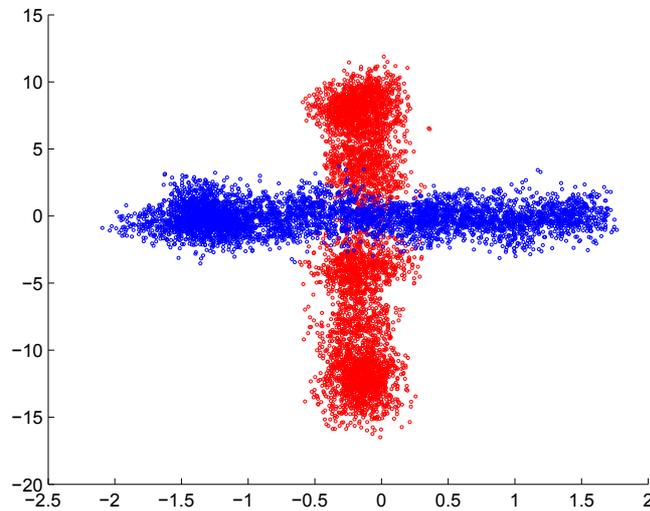


Figure 2.3 : CSP Example.

Other variations to CSP algorithm was proposed, like the Common Spatio-Spectral Pattern (CSSP) algorithm [19]. In the CSSP algorithm simple filters (with one delay

tap) are optimized together with the spatial filters. Recently, a further improvement to the CSSP was presented and called Common Sparse Spectral Spatial Pattern (CSSSP) [20]. This method allows simultaneous optimization of an arbitrary FIR filter within CSP analysis.

A new alternative method based on Sub-band CSP (SBCSP) and score fusion is proposed in [17], where instead of temporal FIR filtering, the EEG signal is decomposed into sub-bands using a filter bank, then CSP is performed in each sub-band and subsequently a sub-band score is defined. The final decision is derived from fusion of the scores from each sub-band.

2.5 Support Vector Machine (SVM)

Support Vector Machines are a very useful and popular technique for data classification, it is based on supervised learning models with associated learning algorithms that analyze data and recognize patterns [21]. Usually, a classification task involves separating data into training and testing sets. In the training set, each instance contains one target value and several attributes. The goal of SVM is to produce a model based on the training data which predicts the target values of the test data given only the test data attributes [22]. In addition to performing linear classification, SVMs can efficiently perform non-linear classification using the kernel trick by implicitly mapping their inputs into high-dimensional feature spaces. In other words, SVM is a supervised learning algorithm that classifies linear and nonlinear data based on maximizing margin between support points and a nonlinear mapping to transform the original training data into higher dimensions [23].

Given a training set of instance, label pairs $(x_i, y_i), i = 1, \dots, l$ where l is the number of classes, the support vector machines, according to [24], require the solution of the following optimization problem:

$$\begin{aligned}
& \min_{\alpha_i} \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j K(x_i, x_j) \\
& \text{subject to } \sum_{i=1}^l \alpha_i y_i = 0 \text{ and } 0 \leq \alpha_i \leq C
\end{aligned} \tag{2.9}$$

where C is the penalty factor of the error term, α_i are Lagrange multipliers which represents the direction of the optimal hyperplane, and $K(x_i, x_j)$ is the kernel function. Basically, there are four kernel functions:

- Linear: $K(x_i, x_j) = x_i^T x_j$.
- Polynomial: $K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0$.
- Radial Basis Function (RBF): $K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \gamma > 0$.
- Sigmoid: $K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r)$

where γ, d , and r are kernel parameters. For implementing SVM, each data instance should be represented as a vector of real numbers, this implies that if there are categorical attributes, it has to be converted into numeric data. Then, a scaling process before applying SVM is very important to avoid attributes in larger numeric ranges dominating those in smaller numeric ranges. Another advantage is to avoid numerical difficulties during the calculation, because kernel values depend on the inner products of feature vectors [22]. Normally, each attribute is scaled linearly to the range $[-1, +1]$ or $[0, 1]$, of course, the same method for scale training and testing data is required.

After successful data preprocessing has been done, the next step is to select a kernel. In general, the RBF kernel is a reasonable first choice, because it can handle the case when the relation between class labels and attributes is nonlinear, also, the RBF

kernel has fewer numerical difficulties because it has less hyper parameters.

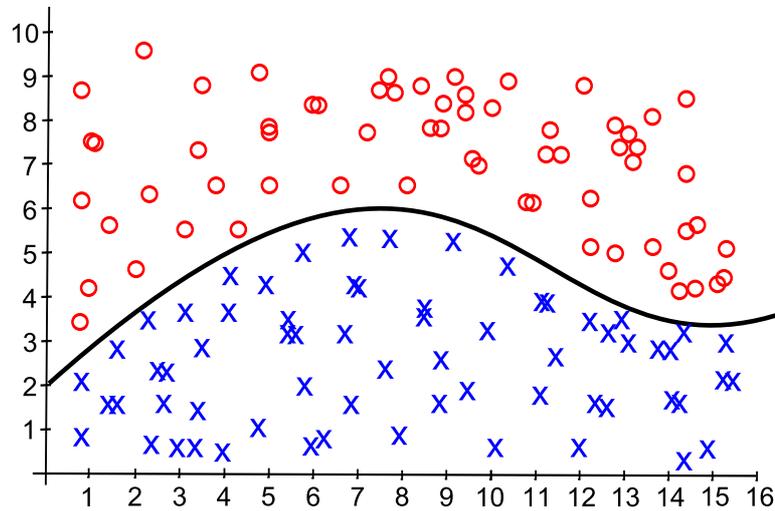


Figure 2.4 : SVM Mapping Example.

An example of SVM mapping is shown in Figure 2.4, the blue crosses indicate data points that belong to category 1 and the red circles that represent data points that belong to category 2. Each of the individual data points has unique input 1 value (x-axis) and a unique input 2 value (y-axis) and all of these points have been mapped to the 2-dimensional space. The support vector machine observes the data in 2 dimensional space, and uses a regression algorithm to find a 1 dimensional hyperplane that most accurately separate the data into its two categories. This separating line is then used by the support vector machine to classify new data points into either category 1 or category 2.

2.6 Naive-Bayes

The Naive Bayes is a classification method based on Bayes Theorem with strong independence assumptions (Naive). In simple terms according to [25], a naive Bayes

classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature.

Depending on the precise nature of the probability model, naive Bayes classifiers can be trained very efficiently in a supervised learning setting. In many practical applications, parameter estimation for naive Bayes models uses the method of maximum likelihood; An advantage of the naive Bayes classifier is that it requires a small amount of training data to estimate the parameters (means and variances of the variables) necessary for classification. Because independent variables are assumed, only the variances of the variables for each class need to be determined and not the entire covariance matrix [26].

To explain the algorithm according to [25], let's assume C_j the class of vector X as belonging to the j -th class, $j = 1, 2, \dots, J$ out of J possible classes. Let $P(C_j|X_1, X_2, \dots, X_P)$ denote the (posterior) probability of the sample vector belonging in the j -th class given the individual characteristics X_1, X_2, \dots, X_P . Furthermore, let $P(X_1, X_2, \dots, X_P|C_j)$ denote the probability of a sample with individual characteristics X_1, X_2, \dots, X_P belonging to the j -th class and $P(C_j)$ denote the unconditional (i.e. without regard to individual characteristics) prior probability of belonging to the j -th class. For a total of J classes, Bayes theorem gives us the following probability rule for calculating the case-specific probability of a sample vector falling in the j -th class:

$$P(C_j|X_1, X_2, \dots, X_p) = \frac{P(X_1, X_2, \dots, X_p|C_j)P(C_j)}{P(X_1, X_2, \dots, X_p|C_1)P(C_1) + \dots + P(X_1, X_2, \dots, X_p|C_j)P(C_j)} \quad (2.10)$$

When applying a trained Bayes classifier to an independent data set it could likely be the case that some of the cases (X_1, X_2, \dots, X_p) that occur in the independent data set do not appear in the training data set. In this case the Naive concept is applied to Equation 2.10, by assuming the inputs (X_1, X_2, \dots, X_p) independent to

each other. This independence allow to calculate the case-specific class probabilities as:

$$P(X_1, X_2, \dots, X_p | C_j) = P(X_1 | C_j) P(X_2 | C_j) \dots P(X_p | C_j) \quad (2.11)$$

The terms on the right-hand-side of the above equation can be calculated simply as the relative frequencies of the individual X_i in the class C_j

2.7 Neural Networks

A neural networks consists of a number of neurons, which are interconnected in often complex ways and organized into layers. A commonly used model of a neuron is called a sigmoidal. In figure 2.5 the structure of this neuronal mode is shown [27].

The model consist of two functional blocks:

1. A linear combiner, which itself consist of a set of weights connected to input terminals. The linear combiner also includes a bias denoted by b_i , which may have a positive or negative value.
2. An activation function, which is the subsequent step to linear combiner. The activation function, denoted by φ , is both nonlinear and memoryless.

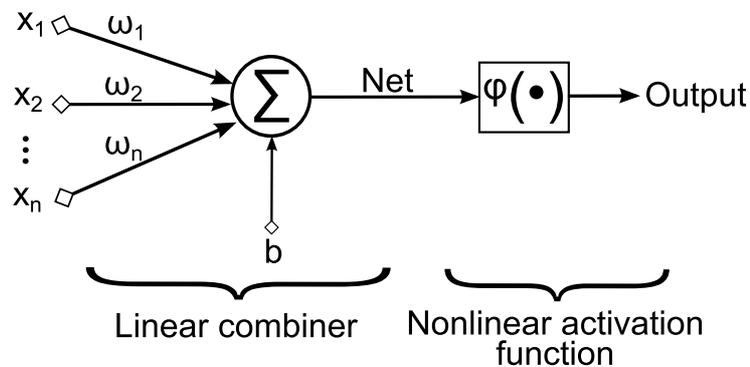


Figure 2.5 : Model of a neuron.

The neural networks are used in three main ways [28]:

- As models of biological nervous systems and intelligence.
- As real-time adaptive signal processors or controllers implemented in hardware for applications such as robots.
- As data analytic methods.

The attempt to resemble biological nervous systems encouraged the development of artificial neural networks by linking numerous simple elements into a highly interconnected system and expecting that as result of self-organization or learning, complex phenomena such as "intelligence" would originate. Artificial neural networks learn in much the same way that many statistical algorithms do estimation, but usually much more slowly than statistical algorithms.

2.8 Hidden Markov Model

To explain the Hidden Markov Model, let us introduce the Markov process. Consider a probability space (Ω, ζ, P) where Ω is the set of all possible outcomes, ζ is the set of all possible events, and $P(A)$ is the probability of $A \in \zeta$. Given a random process $X : \Omega \rightarrow \mathfrak{R}$ and its collection of random variables indexed through time: $X^n = (X_1, \dots, X_n)$.

The chain rule of probability says that for n events $A_1, \dots, A_n \in \zeta$ [29]:

$$\begin{aligned}
 P(A_1 \cap A_2) &= P(A_1)P(A_2|A_1) \\
 P(A_1 \cap A_2 \cap A_3) &= P(A_1 \cap A_2)P(A_3|A_2 \cap A_1) = P(A_1)P(A_2|A_1)P(A_3|A_2 \cap A_1) \\
 &\dots\dots \\
 P(A_1 \cap A_2 \dots \cap A_n) &= P(A_1) \prod_{i=2}^n P(A_i|A^{i-1})
 \end{aligned}
 \tag{2.12}$$

Where $A_{i-1} = A_1 \cap A_2 \dots \cap A_{i-1}$. If we let the event A_i pertain to the event $X_i = a_i$, then we have that [29]:

$$P(X^n = a^n) = P(X_1 = a_1) \prod_{i=2}^n P(X_i = a_i | X^{i-1} = a^{i-1}) \quad (2.13)$$

The above equation shows a causal decomposition of the joint distribution on X^n into a product of conditional distributions of the present value of X given the past values. In conclusion X^n is a *Markov Process* if the future X_{i+1} is independent of the past X^{i-1} given the present X_i [29]:

$$P(X_{i+1} = a_{i+1} | X_i = a_i, X_{i-1} = a_{i-1}) = P(X_{i+1} = a_{i+1} | X_i = a_i) \quad (2.14)$$

Now, suppose that X^n describes a process and it cannot be observed. Instead, a random process Y^n is observable and is statistically linked to X^n . This is the Hidden Markov Model scenario. According to [29], the coupling of Y^n to X^n can be described according to the following definition:

$$P(Y_i = b_i | Y^{i-1} = b^{i-1}, X^n = a^n) = P(Y_i = b_i | X_i = a_i) \quad (2.15)$$

This is a fundamental assumption for hidden Markov models. It basically means that " Y_i is a noisy version of only X_i ". HMM have found greatest use in problems like speech or gesture recognition.

Chapter 3

METHODOLOGY

Signal classification can be described in a very broad way using a flow chart detailing the digital signal processing steps and procedures needed to achieve satisfactory results. In digital signal processing there is a generic flowchart, shown in figure 3.1, that describes the process of signal classification. The first step of the process is signal acquisition, where the signals are to be recorded from the sensors; a stage called preprocessing in which the irrelevant information, like noise, is removed from the signals. Also in this step, the signal is transformed for easier manipulation of the data. The next stage converts the signal to data that contains relevant information which can be interpreted by the last stage called classification. In this last step, a model or network is trained to allow the identification of specific signals.

For EEG signals the key-points are:

- Normally, these signals are very noisy, it is important to have good noise reduction.
- Since there will exist a best frequency band for classification, a good frequency band selection is required.
- Depending on the EEG application, there are more important brain zones than others so the channel selection (spatial filter) is important.

In the first stage, the signals from nature or artificial environments are recorded by using sensors. For some kind of signals, this stage could be the most difficult stage in all the digital signal process, principally when the signal has low SNR. This work is

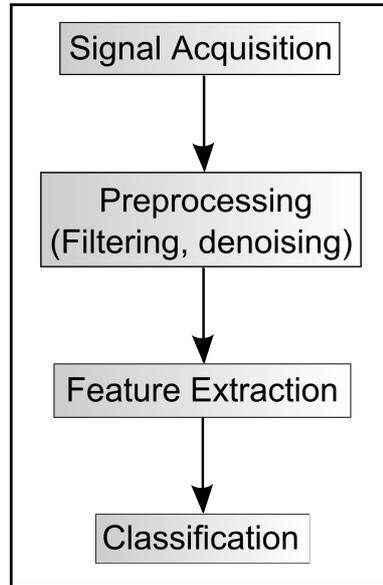


Figure 3.1 : Methodology Flowchart

focused in motor imagery EEG signals which is the electrical signal produced in the subject's brain during the imagination of motor actions.

3.1 Signal Acquisition

The method used for signal capture is non-invasive. Invasive methods provide clearer more precise signals but require medical procedures that increase the level of risk for the subjects. At this initial stage of research it is easier to test the broad hypothesis before proceeding to more elaborate signal capture schemes. In the follow sections the hardware, software and the experimental approach is described.

3.1.1 Hardware

Due to specifications of the hardware used in chapter 4 (Emotiv EPOC Neuro-headset). which limited the quality of the captured signal, a different hardware was used to capture the EEG signals for the final testing stages. The selected hardware system consists of: BrainAmp and EasyCap from Brain Products. EasyCap is an EEG Recording Cap, which is worn on the head by the subject and captures 32 channels through ring-shaped electrodes with highly efficient impedance minimization. The

position of the electrodes in the subjects scalp is shown in Figure 3.2, the electrodes in the sensorimotor cortex area (black highlighted) are used in the experiment to measure the ERD.

One of the main advantages of the Brain Products system is that instead of using a wireless communication interface to send the signals from the headset to the computer, it uses twin fiber optics, making the communication more robust against noise. It also has a higher sampling frequency and therefore better resolution in the signal acquisition. The number of electrodes is 32, which translates in greater cranial coverage so more brain zones can be monitored.

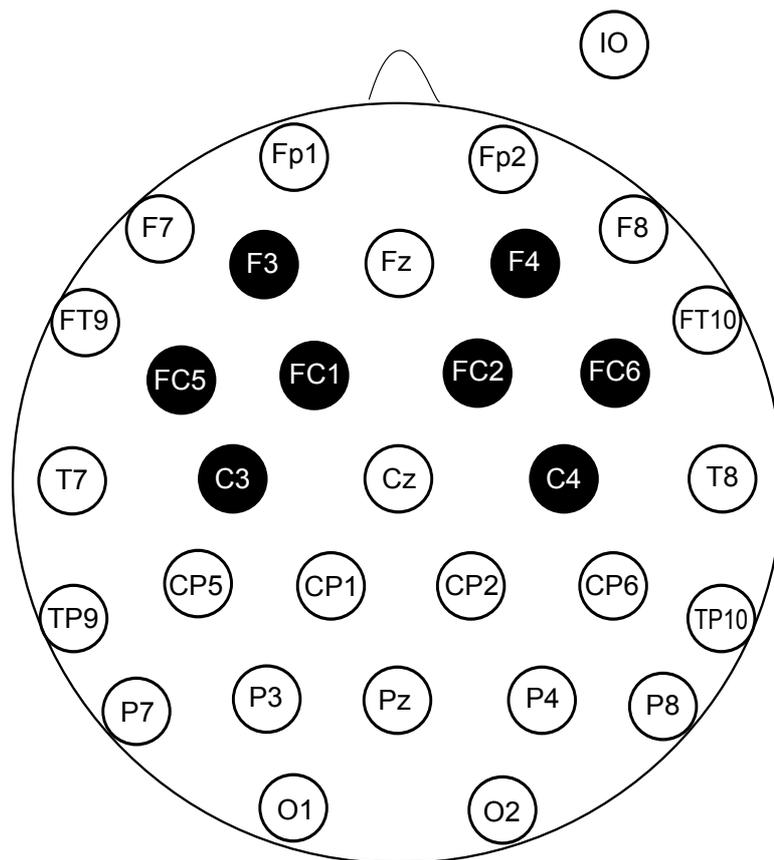


Figure 3.2 : Brain Map and channel selection.

The signal captured by Easycaps electrodes is handed to BrainAmp which is an electronic device in charge of amplifying, treating and digitalizing the data acquired

by the EEG cap, see Figure 3.3 . BrainAmp includes a sequential 16 bit analog to digital converter, with a 5 KHz sample rate per channel, and a filter bank that limits the signals to those with frequencies between 0.016 Hz and 1 KHz. Its technical specification is given in Table 3.1 .



Figure 3.3 : BrainAmp and Powerpack

To reduce the electrode impedance and obtain more accurate signals, it is necessary to use electrode gel and neurodiagnostic electrode paste; signa and uprep gels were applied in the individual electrodes and Ten20 was used to stick the IO electrode to subjects cheek. A photo of all this product are show in Figure 3.4 .

For the experiments, 8 channels which are positioned near the motor cortex were chosen (F3, F4, FC1, FC2, FC5, FC6, C3, and C4). The signals were sampled at 400 Hz and low pass filtered to 100Hz.

3.1.2 Software

Brain Products provides a software suite called BrainVision with 2 applications as a part of the EEG system. These are BrainRecorder and BrainAnalyzer. The first one provides an interface that allows the user to visualize the state of the impedance of the electrodes as well as monitoring and storing of the EEG signals acquired from the scalp. BrainAnalyzer contains a series of tools and functions that did not offer much use for the research presented, so it was only used to export the EEG signals,

Table 3.1 : BrainAmp's Technical Specifications

Specification	Value
Number of Channels per unit	32
Max. Number of Channels	256
Channel Type / Reference	One electrode as reference
Integrated Impedance Measurement	Including ref. and ground electrode
Input Impedance for DC	$10M\Omega$
Input Noise	$\leq 2\mu v_{pp}$
CMR	$\geq 90dB$
High Pass	0.016 Hz (10s)
Low Pass	1000 Hz
Operating Range	$\pm 3.28mV$
Resolution	$0.1\mu V$ per bit
Sampling Rate	5 kHz per channel
A/D-Conversion	16 bit
DC Offset Compatibility	$\pm 300mV$
Signal Transmission	Optical (via twin fiber optics)
Power Supply	Rechargeable Battery
Power Consumption	$\leq 150mA$, typical $7mA$ in standby
Computer Interface	USB-Adapter (BUA, dualBUA)
TTL Trigger Input	16 bit
Dimensions H x W x D	$68mm \times 160mm \times 187mm$
Weight	1.1 kg

converting them from a native .eeg file extension to .dat, which allows the signals to be processed and analyzed in Matlab.

The data acquisition, processing and classification of the motor imagery signals was done by using an application written in C++ called EEGClassifier. A more deeply description of this program is covered in the subsection 3.5.

3.2 Signal Preprocessing

Signals in nature are corrupted by noise and signals from other sources apart from the targeted one. These additional components can be seen as irrelevant data mixed in with the desired information, and cleaning the signal in order to remove the excess data can simplify and speed up the following processing stages.



Figure 3.4 : Kit to reduce electrodes impedance

According to [30], brain rhythms -which reflects functional states of different neuronal cortical networks- are blocked by movements, independent of their active, passive or reflexive origin; which are visible bilaterally but pronounced contralaterally in the cortical area that corresponds to the moved limb [20]. This attenuation of the brain rhythms is called ERD, this can be observed over sensorimotor cortex as μ -rhythm which can be measured in the subject's scalp. Since the focal ERD in the motor and sensory cortex can be observed even when a subject is only imagining a movement or sensation in the specific limb, it is important for the BCI to find the topography of the attenuation of the μ -rhythm.

The easycap contains electrodes that measure a voltage, each at a specific site of the subject's scalp. The maximum frequency in which the voltage is sampled at each electrode is 5 KHz. By using an application coded in C++ (EEGClassifier), the sampling frequency was fixed to 400 Hz. The voltage measured by the electrodes are

amplified by a low noise amplifier, low-pass filtered to 100 Hz and quadrature down-converted from a center frequency known to have a narrow-band activity during motor imagery called (μ -frequency). This process is represented in a flowchart shown in the Figure 3.5 .

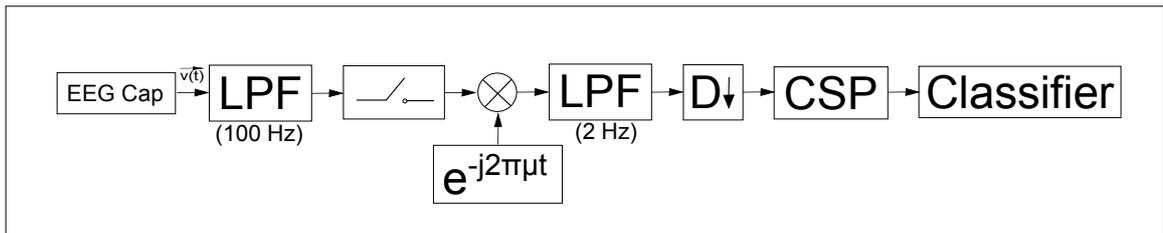


Figure 3.5 : Methodology Diagram.

3.3 Signal Feature Extraction

After cleaning and extracting relevant information from signals, a transformation of the input data into a set of features that can be understandable to the following stages is required. Such transformation should be a reduced representation of data with discriminative information from the input in order to perform a desired task. For each subject, an initial calibration is required before the feature extraction process since the strength of the sensorimotor idle rhythms is known to vary strongly between the subjects. The calibration consist in finding the μ -rhythm which is near to 12 Hz, by collecting EEG data from a subject and looking for the maximum peak of energy between 5 and 25 Hz.

To extract relevant information from Motor Imagery EEG signals, a modified version of CSP, called Common Analytic Spatial Pattern from [1] was used. To understand the concept, let's consider an independent set of electrical signals sources coming from the brain during an EEG activity, which are primarily synchronized spiking among many neurons in specific brain areas with intensities, $\vec{y}(t) = [y_1(t), y_2(t), y_3(t), \dots, y_N(t)]^T$. These source regions are superimposed by passing through a linear mixing channel,

$\mathbf{A} \in R^{N \times N}$ -where N is the number of sources-, composed of surface and volume conduction to create a set of voltages, $\vec{v}(t)$, recorded by the electrodes. Each source oscillates at the same frequency (μHz) but has independent random walk phase, $\vec{\theta}(t) = [\theta_1(t), \theta_2(t), \dots, \theta_N(t)]^T$ so that:

$$\vec{v}(t) = \mathbf{A} \begin{pmatrix} s_1(t) = e^{y_1(t) + j(2\pi\mu)t + \theta_1(t)} \\ s_2(t) = e^{y_2(t) + j(2\pi\mu)t + \theta_2(t)} \\ \vdots \\ s_N(t) = e^{y_N(t) + j(2\pi\mu)t + \theta_N(t)} \end{pmatrix} \quad (3.1)$$

$\vec{v}(t)$ is the voltage measured by the electrodes, it is shown in the figure 3.5 . After filtering, sampling (to convert the analog signal in time domain to a discrete signal) and applying a method called digital down-converted, which converts a digitized real signal centered at an intermediate frequency, in this case μ Hz, to a basebanded complex signal centered at zero frequency, we have:

$$\vec{w}_i = \mathbf{A} \begin{pmatrix} e^{y_{1i} + j\theta_{1i}} \\ e^{y_{2i} + j\theta_{2i}} \\ \vdots \\ e^{y_{Ni} + j\theta_{Ni}} \end{pmatrix} \quad (3.2)$$

where w_i is the sampled signal which feeds the CSP algorithm. The following weighted moving average model was proposed to relate the intensities \vec{y}_i with the type of subject motor imagery performed by a subject over the last J samples times.

$$\vec{y}_i = \sum_{j=0}^{J-1} \vec{\alpha}_j x_{i-j} + \vec{n}_i \quad (3.3)$$

Where x_{i-j} denotes the ideal subject motor imagery at time $i - j$ (-1 or 1 for Left and Right class respectively) relative to current time i , α_j are the weight coefficients, and \vec{n}_i is a vector of additive Gaussian noise. The goal is then to retrieve the class of

motor imagery, x_i , using the values of \vec{w}_i . To achieve this goal, the matrix \mathbf{A} must be estimated. The equation 3.2 can be factored into a diagonal intensity matrix (Φ_i) and a phase vector (\vec{p}_i):

$$\vec{w}_i = \mathbf{A}\Phi_i\vec{p}_i = \mathbf{A} \begin{pmatrix} e^{y_{1i}} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & e^{y_{Ni}} \end{pmatrix} \quad (3.4)$$

The source intensities are considered constant when \vec{w}_i is recorded during training and the type of motor imagery is known and consistent for more than J samples. This implies a constant Φ_i . This diagonal matrix is named Φ_L for constant left motor imagery and Φ_R for constant right motor imagery. Then \vec{w}_i can be written as:

$$W_L = A\Phi_L P \quad (3.5)$$

$$W_R = A\Phi_R P \quad (3.6)$$

The objective is to find \mathbf{A} since P is made up of random phase vectors that are statistically equal in either case. Now, by using the ESPRIT method [31], a matrix $\mathbf{T} \in C^{N \times N}$ is found such that:

$$TW_L = W_R \quad (3.7)$$

Because P has full row rank, and using the equations 3.5 and 3.6:

$$TA = A \frac{\Phi_R}{\Phi_L} \quad (3.8)$$

The fraction $\frac{\Phi_R}{\Phi_L}$ is a diagonal matrix of elements $e^{y_{kR}-y_{kL}}$. Clearly, an element that is very small or very large because source intensities are large ($y_k > 0$) during right motor imagery ($x_i = 1$) and small ($y_k < 0$) during left motor imagery ($x_i = -1$). T is easily computed as $T = W_R W_L'$ for which eigenvectors and eigenvalues can be found.

Using the Common Space Pattern (CSP) concept. Let's consider the covariance matrix of \vec{w}_i made up of $r_{(m,n)} = \langle w_m, w_n \rangle_E$:

$$\begin{aligned} R_{\vec{w}} &= E[\vec{w}_i \vec{w}_i^H] \\ R_{\vec{w}} &= E[(A\Phi\vec{p}_i + \vec{n})(A\Phi\vec{p}_i + \vec{n})^H] \\ R_{\vec{w}} &= A\Phi R_{\vec{p}} \Phi^H A^H + R_{\vec{n}} \\ R_{\vec{w}} &= A\Phi^2 A^H + R_{\vec{n}} \end{aligned}$$

where the terms of the form $E[\vec{p}_i \vec{n}_i]$ are zero because they are uncorrelated between them, $R_{\vec{p}} = I$ because $r_{(m,n)} = \langle p_m, p_n \rangle_E = 0 \forall m \neq n$ from independence. For each motor imagery class, the equation becomes:

$$R_{\vec{w}_L} = A\Phi_L^2 A^H + R_{\vec{n}} \quad (3.9)$$

$$R_{\vec{w}_R} = A\Phi_R^2 A^H + R_{\vec{n}} \quad (3.10)$$

Multiplying by an inverse matrix, A^{-1} , on both right-hand sides of each equation, the two combine to form:

$$R_{\vec{w}_L} A^{-1} = R_{\vec{w}_R} A^{-1} \left(\frac{\Phi_L^2}{\Phi_R^2} \right) \quad (3.11)$$

Now the columns of A^{-1} are the eigenvectors and the diagonal matrix $\frac{\Phi_L^2}{\Phi_R^2}$ the eigenvalues of this problem and can be solved directly. This method is called Common Space Analytic Pattern, which recovers the intensities y_i given a voltage $v(\vec{t})$ measured from the electrodes in the sensorimotor cortex. This method is called to be analytical because it uses the topographical properties of the ERD to calculate the matrix channel A , which is composed of surface and volume conduction.

3.4 Classification

In the final stage, a network was created by specifying a training dataset, the objective of the network is to differentiate between the different classes that it was trained.

Since the inverse matrix, A^{-1} , is found using the method proposed by [1], the sources intensities, in equation 3.2 can be recovered by:

$$\vec{y}_i = \log \|A^{-1}\vec{w}_i\| \quad (3.12)$$

The two largest and smallest eigenvector corresponding to the most discriminative intensities of \vec{y}_i are used. The final step is to estimate x_i at each sample time.

Let's denote the subject's unobservable motor imagery intents during the time interval k by $H_k = L, R$. For the next model, a micro-classification [1] has been made at each sample time i and a classification at each time k . The joint probability distribution can be decomposed into

$$P_{H^k, X^i, Y^i}(h^k, x^i, y^i) = \prod_{\tau=1}^k P_H(h_\tau) P_{X^i|H^k}(x^i|h^k) P_{Y^i|H^k, X^i}(y^i|h^k, x^i) \quad (3.13)$$

where the conditional probability of motor imagery given the intent [1] is

$$P_{X^i|H^k}(x^i|h^k) = \prod_{i'=1}^i 1_{x_{i'}=\gamma(h_k)} \quad (3.14)$$

3.4.1 Hidden Markov Model

Hidden Markov Models (HMM) are dynamic classifiers used in a variety of fields, most widely in the field of speech recognition. A HMM is a kind of probabilistic automaton that can provide the probability of observing a given sequence of feature vectors. A HMM involves probabilities for transition between the states, as well as

conditional probability densities. HMM have been applied to the classification of 2-class temporal sequences of BCI features.

The goal with HMM is to calculate the posteriori probability of a subject's intent to be the left motor imagery class given an intensity y_i ($P(H_k = L|Y^i = y^i)$) by computing the likelihood of seeing the provisional CSAP output given the underlying subject's intent, $P_{Y^i|H^k, X^i}(y^i|h^k, x^i)$. In [1], a HMM assumption is proposed:

$$P_{Y^i|H^k, X^i}(y^i|h^k, x^i) = \prod_{i'=1}^i fw(y_{i'} - \sum_{j=0}^{J-1} \vec{\alpha}_j x_{i'} - j) \quad (3.15)$$

Where $fw(\xi) = (0, \Sigma^2; \xi)$, are calculated using the maximum-likelihood estimation of the parameters $(\vec{\alpha}_1, \dots, \vec{\alpha}_J, \Sigma^2)$, J is based on training data sets and is chosen according to the minimum description length criterion [32].

The Forney factor graph [33] to decompose the joint probability distribution in a graph is used in [1] in order to optimize the classification process. At every step i , a level for y_i and h_k will be added onto the graph. Using the method described before the posterior probability $P(H = L|Y)$ is estimated, a decision is made when its value crosses some threshold τ and a new hypothesis node $P(H_{k+1})$ is added to the graph. Before any decision is made, it is assumed that the graph depends on recent history of y_i and $x_i = x_{i-1} = \dots = x_{T_{k-1}}$.

The HMM model proposed by [1] uses eigenvalue calculation to compute a new CSAP A^{-1} matrix and an updated Least Squares solution to find the $\vec{\alpha}$ and Σ^2 parameters for each cycle using the last as training. The threshold τ used was 0.99999.

3.4.2 Support Vector Machine

Since the intensities y_k can be retrieved using the equation 3.12, knowing the user's intents H_k and the fact that some sources intensities are small during left motor imagery and large during right motor imagery, a support vector machine method could exploit statistically the intensity changes. According to previous works based on the

comparison of efficiency between different kernels applied to EEG signals, the kernel chosen was Radial Basis Function which is:

$$RBF = K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \quad (3.16)$$

Where $\gamma = 5$ and the stopping criteria $\epsilon = 0.00001$. The C++ library used for SVM classifier is called LIBSVM [34].

3.5 Implementation

In order to integrate the data acquisition, preprocessing, the CSAP algorithm and the classifiers, a program in C++ called '*EEGClassifier*' was developed. This application has the following responsibilities:

- Read the configuration file (conf.ini) with the parameters for hardware, protocol and Classifier.
- Generate randomly visual stimuli (left or right) to guide the subject through the training process.
- Synchronize the visual stimuli with the data acquisition.
- Preprocess and filter EEG data.
- Communicate with Matlab for high level math operations.
- Send data to a python graphical user interface to make the application friendly to the user.
- Train and save the classifier.
- Load classifier for posterior use.
- Classify motor imagery data in real time

A general schematic diagram of *EEGClassifier* is shown in the Figure 3.6. Three main stages are visible. The first, pretrain, where the μ frequency is calculated for each subject in the DSP training method. The second stage, the spatial filter w

is calculated and used to filter the data to recover the intensities dW and feed the classifier algorithms. The HMM algorithm used was proposed by [1] and its output are basically three parameters: $alpha$, J and $sigmaInv$; For SVM algorithm [34], the output is a data structure called *model*. In the last stage, called *process* or validation, the μ frequency, the filter w is used to extract information for a single trial signal and using a classifier, estimate its probability to belong to the left class, *pLeft*.

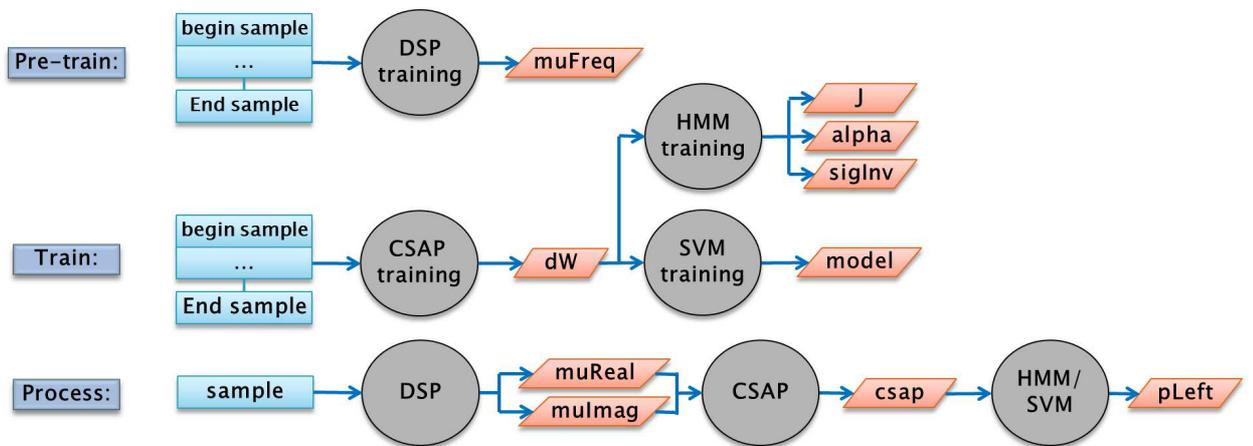
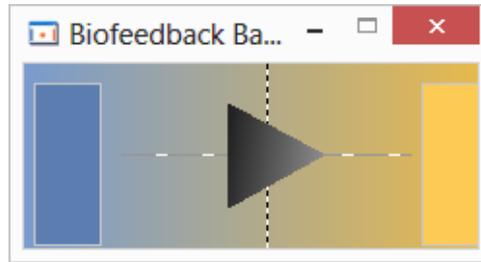


Figure 3.6 : EEGClassifier Schematic diagram.

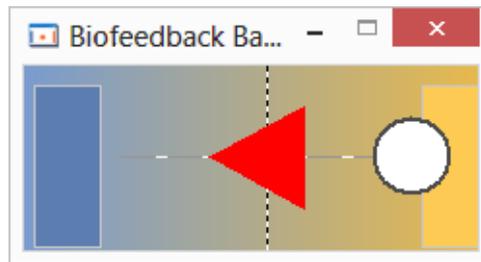
EEGClassifier does not have a Graphical User Interface, instead of this, the application sends data via socket to a Python code which contains a graphical interface that shows information necessary to guide the subject through the training phase using arrows, and another one to test the classifier by using a path speller concept [35].

3.5.1 Ball display

Ball display is a graphical user interface implemented in Python which principal objective is to guide a subject through the training process. This interface receives data from *EEGClassifier* using data socket UDP. The graphical interface for subject's training phase is shown in figure 3.7 (a), a posterior stage tests the classifiers using a visual feedback in the same GUI in Fig. 3.7 (b), where an arrow displays a direction proposed and the ball shows the user's intent.



(a) Python's GUI for training phase



(b) Python's GUI with visual feedback

Figure 3.7 : Visual stimuli used in Training phase

3.5.2 Brain-Computer Interface: Pathspeller

Path speller is basically an interface for tracing smooth planar curves, where input is taken from a EEG cap during left and right motor imagery. This BCI was based on [35] and its model proposed is shown in figure 3.8, which waits until the user's intent is clear before taking any action; this made the BCI accuracy higher but not necessarily faster.

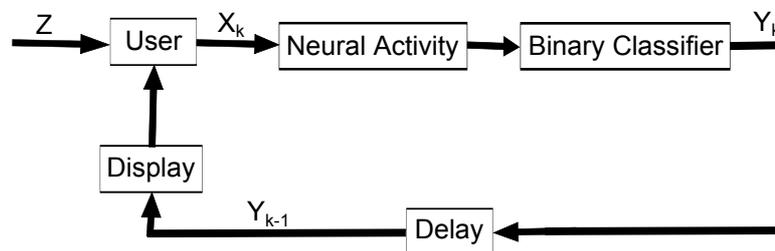


Figure 3.8 : BCI Model.

The importance of this visual interface lies in the feedback, because it allows the user to correct errors made by the BCI in the classification of motor imagery and to

3.6 Experiment

In order to familiarize the subjects with the dynamics of the experiment they were handed a tennis ball to hold in each hand. A direction (left or right) was shown on a screen and they were asked to squeeze the tennis ball in the corresponding hand. The direction was changed every second, working through a predetermined list of ten directions unknown to the subject.

The experiment begins with the pre-training phase, where μ -rhythm is calculated using 30 trials. For signal acquisition, subjects were shown a direction on a screen (5 trials for left and 5 for right generated randomly) and asked to concentrate on the thought of squeezing the tennis ball from the conditioning part of the experiments without squeezing the tennis ball in their hand. They are asked not to squeeze in order to avoid a motor component to the signal and try to guarantee that it is purely brain activity. In the training phase, EEG signal is recorded using arrows as visual stimulus (Figure 3.7 (a)) in a synchronized fashion. Each trial in this stage lasts 1 second, 10 seconds in total. Using the same interface, it is possible to test the accuracy of the classifier in a posterior stage, using the ball display, Figure 3.7 (b). Finally, according to the classifier chosen, some parameters are saved into a DAT file for a posterior use. After trained and tested the chosen classifier, the Pathspeller BCI can be used.

Chapter 4

EXPERIMENTAL RESULTS

In this chapter, time-frequency analysis for feature extraction and classifier methods are used to classify EEG signals. The data used is previously saved (prewritten data) instead of real time data. Two devices were used to capture EEG signals: Emotiv's EPOC Neuroheadset and BrainAmp By Brain Products GmbH.

4.1 Emotiv's EPOC Neuroheadset Experiments

In this experiment the signal were recorded using the Emotivs EPOC Neuroheadset, Research Edition. The headset consists of 14 electrodes following the International 10-20 Location System, including two CMS/DRL references. Figure 4.1 shows the locations covered by the Emotiv Neuroheadset, the letters corresponds to the lobe (frontal, temporal, occipital, etc..), while the number corresponds to the hemisphere; Odd numbers for the left hemisphere and even numbers for the right hemisphere.

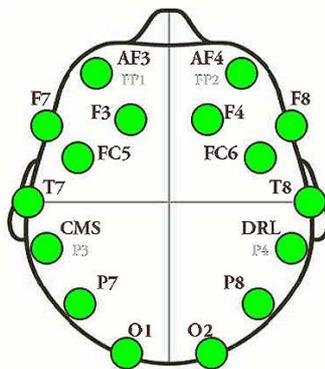


Figure 4.1 : Emotivs EPOC Neuroheadset electrode placement illustration

All subjects were male between the ages of 20 and 30 years. The study consisted of recording the subjects brain activity while performing predefined mental tasks. The recording time was limited to approximately 8 seconds to prevent changes in the frequency content of the signals due to fluctuations in the attention level of the subject. Each task involves the manipulation of the cube showed in figure 4.2 , included in Emotiv software package. The tasks were: Neutral (no manipulation), Pushing and Lifting. During the recording trials, no other visual, auditory or physical stimulus was delivered. The subjects were asked to relax and concentrate only on the mental tasks.



Figure 4.2 : Emotiv Cognitive Suite Cube

In order to successfully analyze and interpret complex multi-dimensional data, dimensionality reduction is done. The purpose of Principal Component Analysis (PCA) is to reduce the number of dimensions, originally 14, to a number that would simplify the pattern recognition process, without significant loss of information. This is accomplished by finding the projections that maximize the variance of the data. The component scores, also called loadings was used in the proposed clustering analysis focusing on the first two component scores.

4.1.1 EEG Classification using SOM

Seven subjects were used to collect data from the EEG, an experiment with a virtual cube was used, and the subjects had to move it (push, lift and leave static). Each subject had two attempts on each activity, and each attempt was recorded. There are two kinds of subjects for this experiment: subjects that had interaction with the virtual cube before (3 subjects), and the beginners (4 subjects). For the neural network training, the data of one subject was chosen, for each activity, principal component analysis were calculated and the first two components were used for training the self-organized map. Then, the input for training has dimensions 2000x3 (3 activities, 1000 samples each principal component). The number of iteration was 100,000; the algorithm takes about 5 minutes for training.

At the start of the classification process all the subjects were treated the same way, the general accuracy for the classification algorithm was 57%, to improve the accuracy, it was necessary to separate the experienced subjects from the beginners. The confusion matrix for the experienced subjects is shown in Table 4.1 . In general, for experienced subjects in this experiment, the classification of EEG signals for the mental task, using SOM was 78%.

In Table 4.2 the confusion matrix for the beginner subjects is shown. The best accuracy was registered for the subject D and E with 67%, Table 4.2 (d) indicates the total accuracy of SOM classification for beginner subject as 61%. Which can be improved by better experimental procedure.

Table 4.1 : SOM Classification Results for experienced subjects

(a) Subject A

	Lift	Neutral	Push	Total	Accuracy
Lift	2	0	0	2	1
Neutral	1	1	0	2	0.5
Push	0	0	2	2	1
Total	3	1	2	6	0.83
Accuracy	0.67	1	1		

(b) Subject B

	Lift	Neutral	Push	Total	Accuracy
Lift	2	0	0	2	1
Neutral	1	1	0	2	0.5
Push	0	0	2	2	1
Total	3	1	2	6	0.83
Accuracy	0.67	1	1		

(c) Subject C

	Lift	Neutral	Push	Total	Accuracy
Lift	2	0	0	2	1
Neutral	0	2	0	2	1
Push	1	1	0	2	0
Total	3	3	0	6	0.67
Accuracy	0.67	0.67	0		

(d) Average for experienced subjects

	Lift	Neutral	Push	Total	Accuracy
Lift	6	0	0	6	1
Neutral	2	4	0	6	0.67
Push	1	1	4	6	0.67
Total	9	5	4	18	0.78
Accuracy	0.67	0.8	1		

4.1.2 EEG Classification using SVM

To test Support Vector Machine classification of EEG Signals, the same subjects and data of the previous section was used, the kernel chosen for SVM was RBF. In the training process, the data of one subject was chosen, for each activity, the principal components were calculated and the first two components were used.

As an example for clustering, a data from experienced subject was chosen, by graphing

Table 4.2 : SOM Classification Results for beginner subjects

(a) Subject D

	Lift	Neutral	Push	Total	Accuracy
Lift	2	0	0	2	1
Neutral	1	1	0	2	0.5
Push	0	1	1	2	0.5
Total	3	2	1	6	0.67
Accuracy	0.67	0.5	1		

(b) Subject E

	Lift	Neutral	Push	Total	Accuracy
Lift	1	0	1	2	0.5
Neutral	1	1	0	2	0.5
Push	0	0	2	2	1
Total	2	1	3	6	0.67
Accuracy	0.5	1	0.67		

(c) Subject F

	Lift	Neutral	Push	Total	Accuracy
Lift	1	1	0	2	0.5
Neutral	1	1	0	2	0.5
Push	1	0	1	2	0.5
Total	3	2	1	6	0.5
Accuracy	0.33	0.5	1		

(d) Average for beginner subjects

	Lift	Neutral	Push	Total	Accuracy
Lift	4	1	1	6	0.67
Neutral	3	3	0	6	0.5
Push	1	1	4	6	0.67
Total	8	5	5	18	0.61
Accuracy	0.5	0.6	1		

the two first PCA of all the mental task (fig. 4.3), we can see two clusters, a group with blue spots representing non activity and the red, representing the activity.

In table 4.3 the results using SVM Classification for experienced subjects is shown, the best accuracy was 67% for subject A. In total, for experienced subjects, the accuracy of this classification algorithm was 61%.

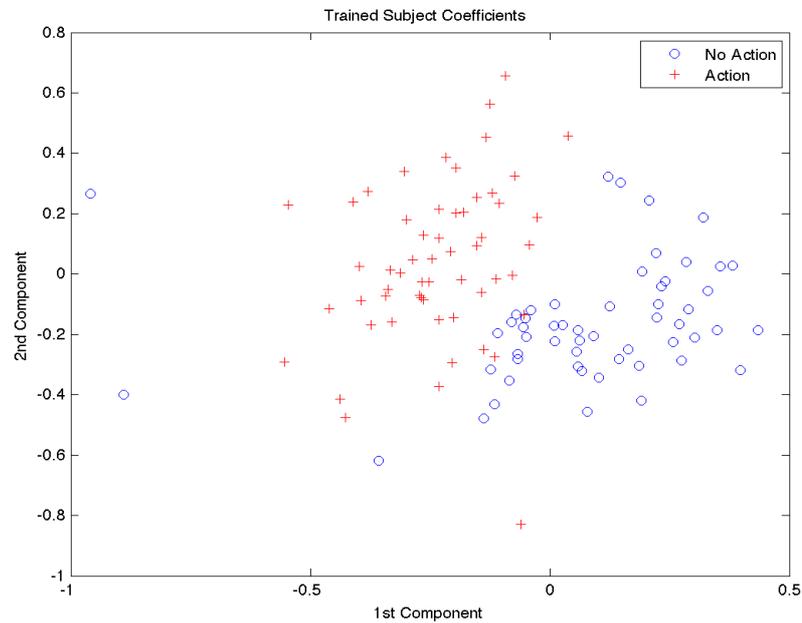


Figure 4.3 : Clustering using PCA

In table 4.4 the results using SVM Classification for beginner subjects is shown, the best accuracy is 67% for subject A. In total, for beginners subjects, the accuracy of this classification algorithm was 50%.

In general, for an optimum classification of the EEG signals recorded in the experiment, it is necessary that the subjects understand the experimental procedure and be acquainted with it before recording the data. By using the SVM algorithm for classification, the general accuracy for mixed subjects was 40%, separating each kind of subject, the classifier improves its accuracy to 61% for experience subjects and 50% for beginners.

In the same way, the accuracy for mixed subjects using the SOM algorithm for classification was 57%, by separating each kind of subject, the classifier improves its accuracy 78% for experienced subjects and 61% for beginners. For this experiment,

Table 4.3 : SVM Classification Results for experienced subjects

(a) Subject A

	Lift	Neutral	Push	Total	Accuracy
Lift	2	0	0	2	1
Neutral	1	1	0	2	0.5
Push	1	0	1	2	0.5
Total	4	1	1	6	0.67
Accuracy	0.5	1	1		

(b) Subject B

	Lift	Neutral	Push	Total	Accuracy
Lift	2	0	0	2	1
Neutral	1	1	0	2	0.5
Push	1	0	1	2	0.5
Total	4	1	1	6	0.67
Accuracy	0.5	1	1		

(c) Subject C

	Lift	Neutral	Push	Total	Accuracy
Lift	2	0	0	2	1
Neutral	1	1	0	2	0.5
Push	1	1	0	2	0
Total	4	2	0	6	0.5
Accuracy	0.5	0.5	0		

(d) Average for experienced subjects

	Lift	Neutral	Push	Total	Accuracy
Lift	6	0	0	6	1
Neutral	3	3	0	6	0.5
Push	3	1	2	6	0.33
Total	12	4	2	18	0.61
Accuracy	0.5	0.75	1		

in conclusion, the Neural Network works better than SVM reaching an accuracy of 78% for experienced subjects.

Table 4.4 : SVM Classification Results for beginner subjects

(a) Subject D

	Lift	Neutral	Push	Total	Accuracy
Lift	2	0	0	2	1
Neutral	1	1	0	2	0.5
Push	1	0	1	2	0.5
Total	4	1	1	6	0.67
Accuracy	0.5	1	1		

(b) Subject E

	Lift	Neutral	Push	Total	Accuracy
Lift	2	0	0	2	1
Neutral	2	0	0	2	0
Push	2	0	0	2	0
Total	6	0	0	6	0.33
Accuracy	0.33	0	0		

(c) Subject F

	Lift	Neutral	Push	Total	Accuracy
Lift	1	1	0	2	0.5
Neutral	1	1	0	2	0.5
Push	1	0	1	2	0.5
Total	3	2	1	6	0.5
Accuracy	0.33	0.5	1		

(d) Average for beginner subjects

	Lift	Neutral	Push	Total	Accuracy
Lift	5	1	0	6	0.83
Neutral	4	2	0	6	0.33
Push	4	0	2	6	0.33
Total	13	3	2	18	0.5
Accuracy	0.38	0.67	1		

4.2 BrainAmp By Brain Products GmbH experiments

4.2.1 Data acquisition and Processing

A graphical interface that generates visual stimuli for the experiments and synchronizes them with the brainamp was used. The application was developed in C++. The application receives an username, number of markers and the time between them as input parameters all given in a text file; there are only two markers, left and right and they are shown randomly and same times each. After showing the directions the application saves separately in text files each channel of the brainamp (from 0 to 31), timestamps and the sequence of markers. Being timestamps the time marker appears in the graphical user interface.

Signals in nature are corrupted by noise and signals from sources aside from the targeted one. These additional components can be seen as irrelevant data mixed in with the desired information, and cleaning the signal in order to remove the excess data can simplify and speed up the following processing stages.

Each electrode measures a voltage at a specific site on the scalp and is sampled at 5 kHz. In figure 4.4 (a) the Fourier transform of electrode C3 for motor imagery from Right hand recorded from a single subject is shown as well its spectrogram, shown in Figure 4.4 (b). From these figures, due to high sampling frequency, non-desirable components are shown, to focus on lower frequencies without losing resolution, downsampling was applied. The EEG signals spectrum and spectrogram resulting are shown in figures 4.4 (c) and 4.4 (d) respectively.

ERD between the most relevant motor imagery information according to [4] are lying μ - and the β - rhythm (12-16 Hz and 18-24 Hz). The EEG signal is easily overwhelmed by activity from artifacts such as muscles or eyes; this presents itself as an excessive level of power on some channels. If the artifact activity is unevenly distributed in the experiments, the CSP will capture it with a high eigenvalue because the CSP

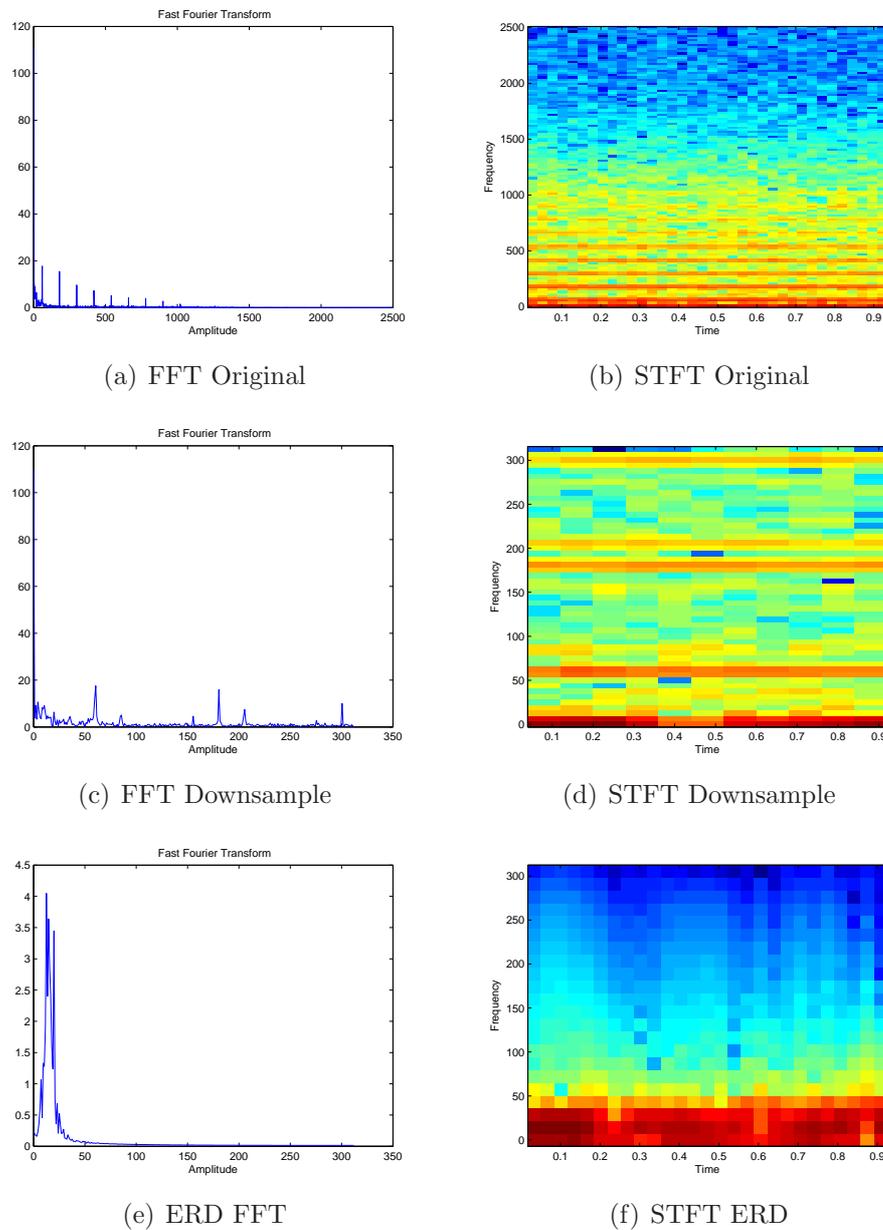


Figure 4.4 : Spectrum and Spectrogram of C3 Electrode using Motor Imagery for Right Hand.

filter pools together the covariance matrix from the trials. This distorts the CSP spatial filter and can be corrected by removing the non-linear contribution from the artifact through empirical mode decomposition. This technique has been used in [36] [37]. Using the Multivariate Empirical Mode Decomposition (MEMD) the ERD from downsampled signals was extracted, the spectrum and spectrogram for channel C3

using right motor imagery are shown in figures 4.4 (e) and 4.4 (f) where filtering can be seen, the resulting signal has frequency components inside the range between μ - and the β - rhythm. This can be done by finding the IMFS corresponding to the desirable frequency range. This is possible due to the fact that each Intrinsic Mode Function has a frequency peak which is found by calculating the maximum amplitude of its spectrum, then an output is created with the sum of those IMFS.

After cleaning and extracting relevant information from signals, a transforming of the input data into a set of features that can be understandable to the following stages is required. Such transformation should be a reduced representation of data with relevant content and discriminative information from the input in order to perform a desired task.

The Common Space Pattern (CSP) is a spatial filter widely used as feature extraction in binary class motor Imagery, An example can be seen in figure 4.5 (a), where red points represents data equivalent to left Motor Imagery and blue for the right Class. As can be appreciated, the outputs from the spatial filter are practically orthogonal between them; this can be used as advantage for the classification stage. Filtering (cleaning) the signal before performing the CSP improves the results, figures 4.5 show the output of the CSP stage for both cases (filtered and un-filtered signal). The cleaned signal produces a more concentrated image than that of the original signal. More importantly, the results from the cleaned signals are centered on x and y axis as shown on various tests with other different samples. Thus by filtering the signal the classifier can reach higher levels of accuracy by priming the signal at each stage. In order to obtain homogeneous CSP data that is easier to classify, larger regions are split to separate the data. The CSP method implemented in Matlab was based on the algorithm from [17] and [18].

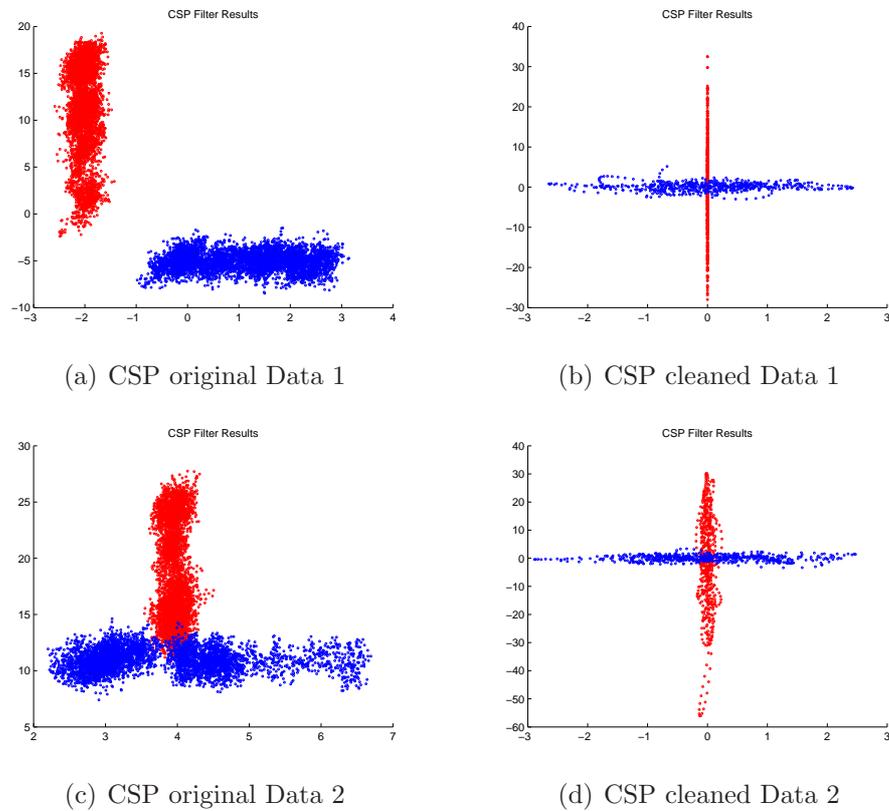


Figure 4.5 : CSP applied to original and cleaned EEG data.

4.2.2 Experimental Results

The application Neurolab was developed in Matlab in order to make signal classification easier and more user friendly. It allows uploading text files containing the EEG data; these text files should have the same format of the output from BCIApp application. When the data is uploaded, the CSP filter is applied automatically and results are shown in a table. The data to train and test the desired classifier can be selected using the information on the table. The application also plots the CSP filter results of the dataset selected by the user.

Neurolab has the following options for classifiers:

- SVM
- Naive-Bayes

For each classifier NeuroLab shows the accuracy, predicted labels, confusion matrix and its ROC curve. Because larger signals take the feature extraction algorithm longer to finish, NeuroLab saves and loads *.mat files with the results of this stage. This allows users to bypass the feature extraction stage for a signal once its *.mat file has been generated and saved. In figure 4.6 the NeuroLab Graphical Interface is shown.

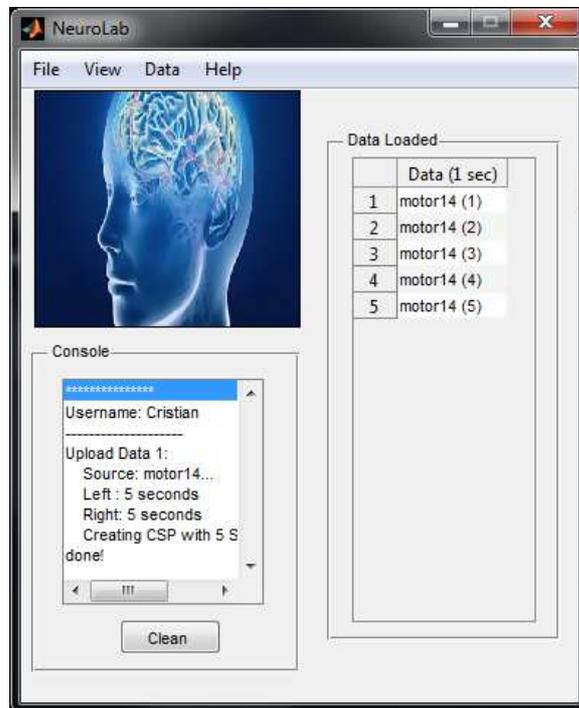


Figure 4.6 : NeuroLab Graphical User Interface

Three subjects (A,B and C) participated in the experiment. The results for SVM classifier are shown in Table 4.5 , where the lowest accuracy (92.4%) was achieved for subject A. The highest accuracy was reached for the subject C (96.3%).

To evaluate the sensibility of the SVM classifier, the ROC curves shown in Figure 4.7 , were drawn using gamma variable from the RBF kernel function as variable threshold.

Table 4.5 : Results using CSP-SVM classifier

(a) Subject A

	Left	Right	Total	Accuracy(%)
Left	72	0	72	100
Right	11	61	72	84.7
Total	83	61	144	
Accuracy(%)	86.7	100		92.4

(b) Subject B

	Left	Right	Total	Accuracy(%)
Left	90	0	90	100
Right	10	80	90	84.7
Total	100	80	180	
Accuracy(%)	90	100		94.4

(c) Subject C

	Left	Right	Total	Accuracy(%)
Left	40	0	40	100
Right	3	37	40	92.5
Total	43	37	80	
Accuracy(%)	93	100		96.3

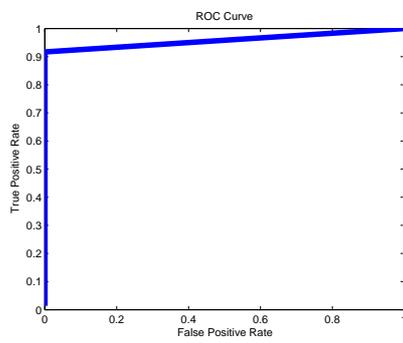
The Naive-Bayes algorithm from Matlab were used to obtain the results shows in Table 4.6 , where the lowest accuracy was achieved by subject A (95.8%). The highest accuracy was reached by the subject B (97.2%).

Using the posterior probabilities, the ROC curves were calculated and are shown in Fig. 4.8 .

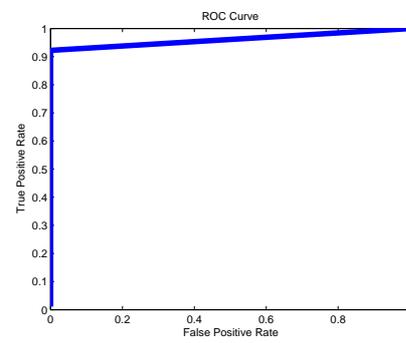
The overall accuracies for SVM and Bayes classifiers are shown in Table 4.7 , where the highest accuracy was reached by Bayes classifier (97.2%) against (93.9%) by SVM classifier.

The ROC curves for SVM and Bayes classifiers are shown in Fig. 4.9 , where an improvement from Naive-Bayes over the SVM can be seen.

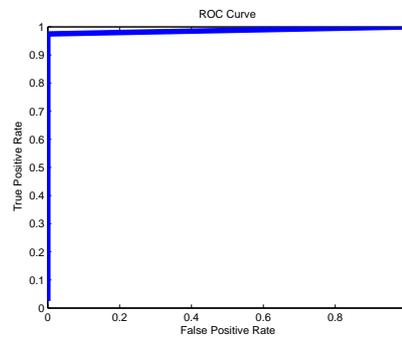
In conclusion of this study using prewritten data, the classifier based on Naive-Bayes method is more accurate than Support Vector Machine with the modified version of the CSP algorithm applied to binary motor imagery EEG signals.



(a) Subject A



(b) Subject B



(c) Subject C

Figure 4.7 : ROC Curves for SVM Classifier.

Table 4.6 : Results using Naive-Bayes classifier

(a) Subject A

	Left	Right	Total	Accuracy(%)
Left	69	3	72	95.8
Right	3	69	72	95.8
Total	72	72	144	
Accuracy(%)	95.8	95.8		95.8

(b) Subject B

	Left	Right	Total	Accuracy(%)
Left	90	0	90	100
Right	5	85	90	94.4
Total	95	85	180	
Accuracy(%)	94.7	100		97.2

(c) Subject C

	Left	Right	Total	Accuracy(%)
Left	37	3	40	92.5
Right	0	40	40	100
Total	37	43	80	
Accuracy(%)	100	93		96.3

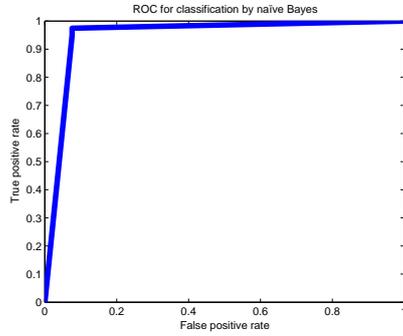
Table 4.7 : Results for all the classifiers

(a) Results using SVM

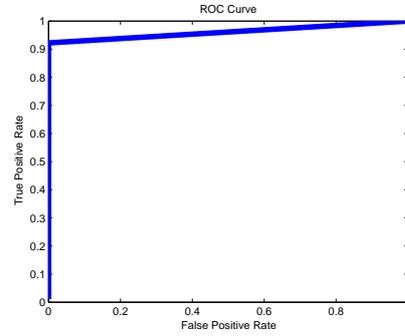
	Left	Right	Total	Accuracy(%)
Left	442	2	444	99.5
Right	52	392	444	88.3
Total	494	394	888	
Accuracy(%)	89.5	99.5		93.9

(b) Results using Naive-Bayes

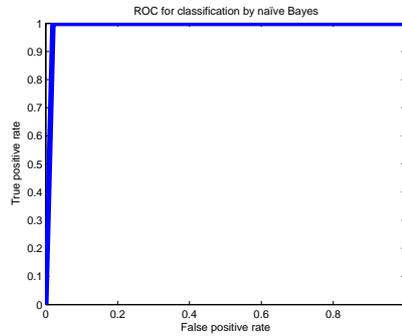
	Left	Right	Total	Accuracy(%)
Left	442	2	444	99.5
Right	20	424	444	94.4
Total	462	446	888	
Accuracy(%)	95.7	99.5		97.2



(a) Subject A

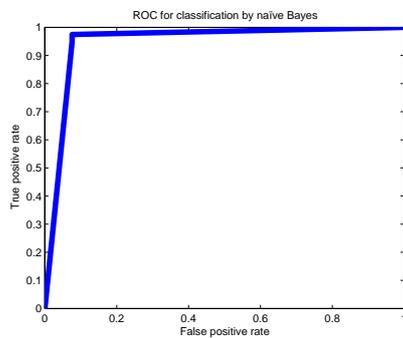


(b) Subject B

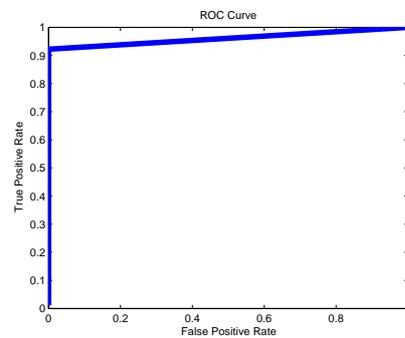


(c) Subject C

Figure 4.8 : ROC Curves for Naive-Bayes Classifier.



(a) SVM Classifier



(b) Naive-Bayes Classifier

Figure 4.9 : ROC Curves comparison for both Classifiers.

Chapter 5

PATH SPELLER RESULTS

After explore different methods for feature extraction and classification for EEG Signals, the Common Space Analytical pattern, Hidden Markov Model and Support Vector Machine were implemented to run a BCI (Pathspeller) in real time due to its low computation time and high accuracy.

5.1 Accuracy

Two subjects, A and B were sitting in a comfortable armchair in front of a computer screen, two sessions on different days were recorded for each subject until they were totally trained with the experiment. The results for subject A is shown in Table 5.1, where an accuracy of 92.5% was obtained for HMM algorithm, with only 3 error in 40 trials. For SVM algorithm the confusion matrix is shown in table 5.1 (b), where the total accuracy was 87.5%, with 5 errors in 40 trials.

For subject B, the results are shown in Table 5.2. The confusion matrix for HMM classifier, Table 5.2 (a), shows a total accuracy of 90%, which means 4 errors in 40 trials. For SVM algorithm, the results can be seen in Table 5.2 (b), where the total accuracy was 85% 6 errors in 40 trials.

5.2 Processing time

The objective of the classifiers used in this work is to reach the probability that a trial belonging to the left class is above a threshold (τ) or the probability of being the right class to be below $(1 - \tau)$. The threshold (τ) for HMM method was fixed to be 0.9999, and for SVM algorithm 0.7. The threshold for SVM is lower than HMM

Table 5.1 : Results for Subject A

(a) HMM results in real time

	Left	Right	Total	Accuracy(%)
Left	20	0	20	100
Right	3	17	20	85
Total	23	17	40	
Accuracy(%)	87	100		92.5

(b) SVM results in real time

	Left	Right	Total	Accuracy(%)
Left	20	0	20	100
Right	5	15	20	75
Total	25	15	40	
Accuracy(%)	80	100		87.5

Table 5.2 : Results for Subject B

(a) HMM results in real time

	Left	Right	Total	Accuracy(%)
Left	19	1	20	95
Right	3	17	20	85
Total	22	18	40	
Accuracy(%)	86.3	94.4		90

(b) SVM results in real time

	Left	Right	Total	Accuracy(%)
Left	20	0	20	100
Right	6	14	20	70
Total	26	14	40	
Accuracy(%)	0.77	100		85

because the SVM classifier tries to match an incoming data with a trained data, while the HMM is a dynamic classifier. After several experiments, the highest threshold reached for SVM classifier is 0.7, this means the highest probability of a trial to match the left class trained is 70%.

When the output of the classifiers are inside the range $[1 - \tau, \tau]$, the result is ignored this makes both classifiers more accurate but with a time cost. For both classifiers, the time spent since the data is taken to the output, no matter if the probability is

inside or outside the range, is between 1 and 3 milliseconds. The time cost for each classifier depends on how quick the confidence level is reached by using the range. In Table 5.3, the timing cost for each classifier in both subjects were measured. In both cases, the HMM algorithm converges faster than SVM to the confidence zone desired.

Table 5.3 : Classification time in milliseconds

(a) Subject A

Class	HMM	SVM
Left	29	1040
Right	35	1066

(b) Subject B

Class	HMM	SVM
Left	32	1022
Right	34	1048

The technical specifications of the computer where experiments have been done are shown in Table 5.4.

Table 5.4 : Computer characteristic

Item	Description
Processor	Intel(R) Xeon(R) x5450
Sockets	2
Cores	8
Logical Processor	8
L1 Cache	512KB
L2 Cache	24MB
Maximum Speed	3 GHz
RAM Memory	8 GB
Operative System	Windows 8 Enterprise (64 Bits)

5.3 Path Speller

A BCI, called path speller [35], was used in order to test the classifiers. Basically the path speller is a graphical interface coded in python which draws a curve, the objective is to follow it by using points. The subject can move the points, one at a

time, until the application decides the final position of the current point based on probability and continuing with the next point. The user can move the point using left or right motor imagery signals, depending on the point's orientation the input could be interpreted as up or down. In figure 5.1 the curve draw for subject A and subject B is shown using HMM classifier.

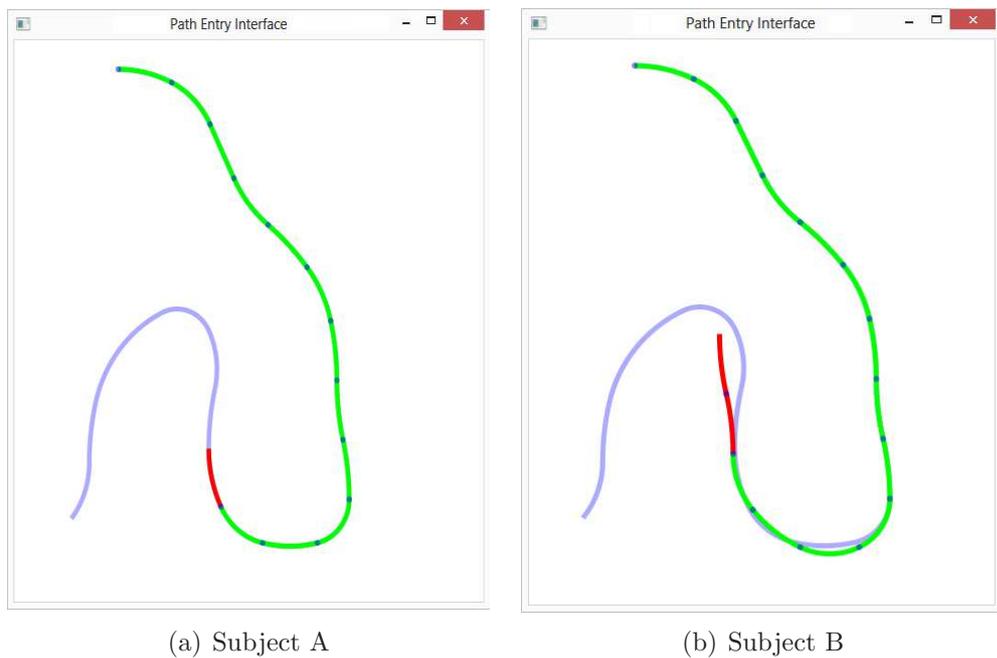


Figure 5.1 : Results of HMM classifier in the Pathspeller BCI.

Figure 5.2 shows the curve drawn by subject A and Subject B using SVM classifier. An improvement of HMM over SVM method can be seen.

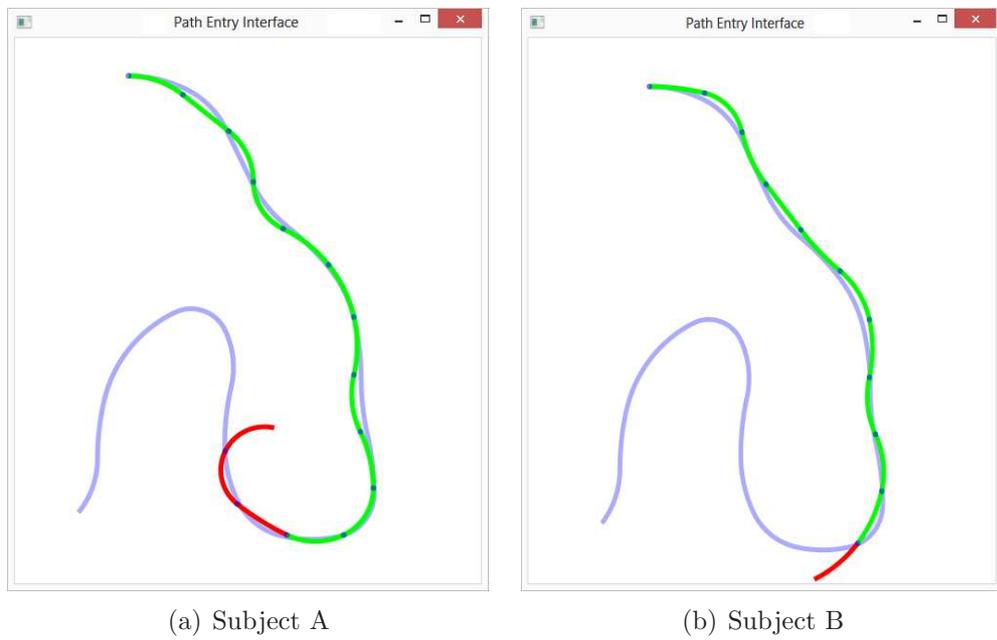


Figure 5.2 : Results of SVM classifier in the Pathspeller BCI.

Chapter 6

CONCLUSIONS

6.1 Contributions

The Common Analytic Space Pattern method was proven to be an efficient algorithm to extract features from EEG signals in experiments based on motor imagery. The source intensities were successfully calculated using this method, which are the basis for the performance of the classifiers.

The SVM algorithm with the approach of the CSAP method was used with outstanding results in binary motor imagery EEG signals and included into the path speller BCI. The HMM classifier reach a very good confidence level with a very good timing. Due to last processing a delay of 1 second was needed before each classification to avoid drawing circles in the Path speller.

The Support Vector Machine was compared with the classifier based on Hidden Markov model in real time using binary motor imagery EEG signals. The HMM performed better than SVM, it was more accurate and faster.

The path speller BCI provides visual feedback which is helpful because it allows the user to correct errors made by the BCI in the classification of motor imagery and to avoid needless redundancy if errors were not made.

6.2 Future Work

The methods presented in this work would provide promising outcomes in the EEG classification area. Future work can examine the possibility of using the methods in the application of the EEG signal classification. To facilitate the further

development of these proposed methods, a few key highlighted issues are addressed below.

- It is possible to extend the proposed binary motor imagery classification to a fourth class version by using the CSAP approach. Since the source intensities can be recovered, a spatial filter could work to distinguish between different classes.
- Exploit the efficiency of the CSAP-HMM algorithm in other BCIs. Its accuracy and speed tested in real time process could be a great contribution in other applications.

Bibliography

- [1] M. McCormick, Rui Ma, and T.P. Coleman. An analytic spatial filter and a hidden markov model for enhanced information transfer rate in eeg-based brain computer interfaces. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 602–605, March 2010.
- [2] Shuren Q.; Zhong J. Extraction of feature information in eeg signal by virtual eeg instrument with the functions fo time-frequency analysis. *Proc. of 6th International Symposium on Image and Signal Processing and Analysis*, pages 7–11, Sept 2009.
- [3] Zhong J.; Shuren Q. Detection of eeg basic rhythm feature by using band relative intensity ratio (brir). *ICASSP*, VI:429–432, Apr 2003.
- [4] G. Pfurtscheller and F. H. Lopes da Silva. Event-related EEG/MEG synchronization and desynchronization: basic principles. *Clinical neurophysiology*, 110(11):1842–1857, November 1999.
- [5] Huang N. E.; Sheng Z.; Long S.; Wu M.; Shih H.; Zheng Q.; Yen N.; Tung C.; Liu H. The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis. *Proc. R. Soc. Lond.*, 454:903–995, Mar 1998.
- [6] Rehman N.; Mandic D. P. Multivariate empirical mode decomposition. *Proc. R. Soc. Lond.*, 466:1291–1302, Dec 2009.
- [7] N. ur Rehman and D.P. Mandic. Empirical mode decomposition for trivariate signals. *Signal Processing, IEEE Transactions on*, 58(3):1059–1068, March 2010.
- [8] N. Rehman and D. P Mandic. Multivariate empirical mode decomposition. *Proc. of the Royal Society*, 466(2117):1291–1302, 2010.
- [9] T. Tanaka and D.P. Mandic. Complex empirical mode decomposition. *Signal Processing Letters, IEEE*, 14(2):101–104, Feb 2007.
- [10] M. Umair Bin Altaf, T. Gautama, T. Tanaka, and D.P. Mandic. Rotation invariant complex empirical mode decomposition. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 3, pages III–1009–III–1012, April 2007.

- [11] G. Rilling, P. Flandrin, P. Goncalves, and J.M. Lilly. Bivariate empirical mode decomposition. *Signal Processing Letters, IEEE*, 14(12):936–939, Dec 2007.
- [12] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2(13):37 – 52, 1987. Proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists.
- [13] Lynne Abdi, Herve; Williams. *Computational Statistics*, volume 2. John Wiley & Sons, Inc., 2010.
- [14] H. Ramoser, J. Muller-Gerking, and G. Pfurtscheller. Optimal spatial filtering of single trial eeg during imagined hand movement. *Rehabilitation Engineering, IEEE Transactions on*, 8(4):441–446, Dec 2000.
- [15] Johannes Mller-Gerking, Gert Pfurtscheller, and Henrik Flyvbjerg. Designing optimal spatial filters for single-trial EEG classification in a movement task. *Clinical neurophysiology*, 110(5):787–798, May 1999.
- [16] G. Dornhege, B. Blankertz, G. Curio, and K. Muller. Boosting bit rates in noninvasive eeg single-trial classifications by feature combination and multiclass paradigms. *Biomedical Engineering, IEEE Transactions on*, 51(6):993–1002, June 2004.
- [17] Q. Novi, Cuntai Guan, T.H. Dat, and Ping Xue. Sub-band common spatial pattern (sbcs) for brain-computer interface. In *Neural Engineering, 2007. CNE '07. 3rd International IEEE/EMBS Conference on*, pages 204–207, May 2007.
- [18] Yijun Wang, Shang-kai Gao, and Xiaorong Gao. Common spatial pattern method for channel selection in motor imagery based brain-computer interface. In *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the*, pages 5392–5395, Jan 2005.
- [19] S. Lemm, B. Blankertz, G. Curio, and K. Muller. Spatio-spectral filters for improving the classification of single trial eeg. *IEEE Trans. Biomed. Eng*, 52:1541–1548, 2005.
- [20] G. Dornhege, B. Blankertz, M. Krauledat, F. Losch, G. Curio, and K. Muller. Optimizing spatio-temporal filters for improving brain-computer interfacing. In *Advances in Neural Inf. Proc. Systems (NIPS 05)*, pages 315–322. MIT Press, 2005.
- [21] Cortes Corinna; Vapnik Vladimir. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995-09-01.

- [22] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin. A practical guide to support vector classification. Technical report, Department of Computer Science, National Taiwan University, 2003.
- [23] David Meyer, Friedrich Leisch, and Kurt Hornik. The support vector machine under test. *Neurocomputing*, 55:169 – 186, 2003. Support Vector Machines.
- [24] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory, COLT '92*, pages 144–152, New York, NY, USA, 1992. ACM.
- [25] Tom Fomby. Naive-bayes classifier. *Southern Methodist University*, pages 1 – 3, 2008.
- [26] F. Klawonn and P. Angelov. Evolving extended naive bayes classifiers. In *Data Mining Workshops, 2006. ICDM Workshops 2006. Sixth IEEE International Conference on*, pages 643–647, Dec 2006.
- [27] Simon Haykin. *Adaptive Filter Theory*. Prentice Hall, forth edition, 2002.
- [28] W.S Sarle. Neural networks and statistical models. *in Proceedings of the Nineteenth Annual SAS Users Group International Conference*, pages 1538 – 1550, 1994.
- [29] Richard Duda; Peter Herve; and David Stork. *Pattern Classification*. John Wiley & Sons, Inc., 2001.
- [30] G. Pfurtscheller and F. H. L da Silva. Event-related eeg/meg synchronization and desynchronization: basic principles. *Clin. Neurophysiol.*, pages 96 – 115, 1999.
- [31] R. Roy and T. Kailath. Esprit-estimation of signal parameters via rotational invariance techniques. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 37(7):984–995, Jul 1989.
- [32] A. Barron, J. Rissanen, and Bin Yu. The minimum description length principle in coding and modeling. *Information Theory, IEEE Transactions on*, 44(6):2743–2760, Oct 1998.
- [33] H.-A. Loeliger. An introduction to factor graphs. *Signal Processing Magazine, IEEE*, 21(1):28–41, Jan 2004.

- [34] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [35] Cyrus Omar, Abdullah Akce, Miles Johnson, Timothy Bretl, Rui Ma, Edward L. Maclin, Martin McCormick, and Todd P. Coleman. A feedback information-theoretic approach to the design of brain-computer interfaces. *Int. J. Hum. Comput. Interaction*, 27(1):5–23, 2011.
- [36] M. K I Molla, T. Tanaka, T.M. Rutkowski, and A. Cichocki. Separation of eeg artifacts from eeg signals using bivariate emd. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 562–565, March 2010.
- [37] Norden E. Huang; Zheng Shen; Steven R. Long; Manli C. Wu; Hsing H. Shih; Quanan Zheng; Nai-Chyuan Yen; Chi Chao Tung and Henry H. Liu. The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis. *Proc. of the Royal Society*, 454(1971):903–995, 1998.