

A Preliminary Analysis of the Brain's Response to Single Sound-Parameter Variations

Antonio Trevisi¹ and Anuhi Ruiz Rivera²

Department of Computer Science, Aalborg University

June 10, 2014

¹atrevi12@student.aau.dk

²aruizr11@student.aau.dk

This page is intentionally left blank.

Title:

A Preliminary Analysis of the Brain's Response to Single Sound-Parameter Variations

Theme:

Brain Computer Interfaces
Machine Learning

Project period:

Machine Intelligence, 10th Semester
Spring of 2014

Project group:

Group mi101f14

Students:

Antonio Trevisi
Anuhi Ruiz Rivera

Supervisors:

Bo Thiesson
Mark Grimshaw

No. of Pages: 128

No. of Appendix Pages: 0

Completed: 9th June 2014

Abstract:

This project attempts to describe the relationship between single sound-parameter variations and the brain's EEG signals by finding a linear regression model that accurately estimates the brain's state given a sound-parameter value. In this project the covariance matrix of each trial is considered a points in Riemannian space. Therefore, the statistical regression analyses use Riemannian geometry and Riemannian distance to find the linear model.

Two sound parameters were selected for the experiments, loudness and frequency. The raw EEG data obtained from the experiments was filtered to remove noise and artifacts. Different regression and cross validation methods were used to find a linear behavior in the data.

The relationship between single sound-parameter variation and the brain's state was found to be subject dependent. The linear models found for each subject validate the use of interpolation to estimate the state of the brain for any loudness or frequency value.

Contents

I	Overview	5
1	Introduction	6
1.1	Project Statement	7
1.2	Structure of the report	9
2	Related Works	10
2.1	Emotion Classification	10
2.2	Sound and Music Theory	11
2.2.1	Frequency and Pitch	11
2.2.2	Timbre, Tone and Note	12
2.2.3	Fundamental Frequency and Harmonics	13
2.2.4	ADSR Envelope	14
2.2.5	Loudness	15
2.2.6	Perception of Sounds	16
2.2.7	Diatonic and Chromatic Scale	18
2.3	Emotional Response to Sound Parameters	18
2.3.1	Loudness	18
2.3.2	Frequency	19
2.3.3	ADSR	19
2.3.4	Sound Source And Localization	19
2.3.5	Sound Effects	20
2.3.6	Signal to Noise Ratio (SNR)	20
2.4	Emotional Response to Music	20
2.4.1	Musical Features	21
2.5	Brain and Emotion	22
3	EEG Technology	24
3.1	Emotiv EPOC	24
3.2	Electrode Placement	25
3.2.1	Brain Rhythms	28
3.3	Notation	29
3.3.1	Channels, Samples and Trials	29

3.3.2	Spatial Covariance Matrix and Sample Covariance Matrix (SCM)	29
II	Theory	31
4	Riemmanian Geometry	32
4.1	Operations on Vector Space	33
4.1.1	Inner Product on a Vector Space	33
4.1.2	Frobenius Norm	33
4.1.3	Vectorizing a Matrix	34
4.2	Differentiable Manifold	35
4.2.1	Riemannian Metric	35
4.2.2	Riemannian Manifold	35
4.2.3	Symmetric Positive Defined Matrices	35
4.2.4	Logarithm and Exponential of a Matrix	35
4.3	Riemannian Distance	36
4.3.1	Logarithmic and Exponential Map	37
4.4	Means of SPD Matrices	38
4.4.1	Arithmetic Mean	38
4.4.2	Geometric Mean	39
4.5	Computing the Mean on the Tangent Space	39
5	Regression	41
5.1	Our Feature Function	43
5.2	Regression on the Feature Transformed Input	44
5.3	Least Square Regression and Ridge Regularization	47
5.4	Dual Representation with Riemannian-Based Kernel	49
5.5	Constructing a Gaussian Kernel	51
6	Estimation Of Covariance Matrices	53
6.1	Interpolation	54
6.1.1	Interpolation in the Euclidean Space	54
6.1.2	Interpolation in the Riemannian Manifold	54
6.1.3	Interpolation In Our Project	55
III	Sound Parameter Variation Analysis	59
	Overview of the Analysis	60
7	Signal Acquisition: Experiments	62
7.1	Experiment Design	63
7.1.1	Loudness Experiment	64
7.1.2	Frequency Experiment	66

7.2	Experiment Implementation	67
7.3	Experiment Execution	67
8	Pre-Processing	70
8.1	Ordering and Cutting the Trials	70
8.1.1	Filtering	71
8.1.2	Transformation of the Target Values	72
9	Data Analysis Implementation	73
9.1	Feature Transformation: Riemannian Space to Tangent Space	74
9.2	Ridge Regression	75
9.3	Dual Representation Regression with Riemannian-based kernel	76
10	Interpolation Implementation	77
IV	Results and Conclusion	79
11	Results	80
11.1	Visual Interpretation of the Results	81
11.2	Cross Validation	81
11.3	Loudness Experiment - All Subjects	84
11.3.1	LOSO Cross Validation, 14 Channels, All Timbres . .	84
11.3.2	LOSO Cross Validation, 1 Channel, All Timbres . . .	85
11.3.3	LOSO Cross Validation, 14 Channels, Separate Timbres	87
11.4	Loudness Experiment - Separate Subjects	88
11.4.1	LOTiO Cross Validation, 14 Channels	88
11.4.2	LOTO Cross Validation, 14 Channels, All Timbres . .	89
11.4.3	LOTO Cross Validation, 8 Channels, All Timbres . . .	92
11.4.4	LOTO Cross Validation, 6 Channels, All Timbres . . .	96
11.5	Separate Subjects - Summary	99
11.6	Frequency Experiment - All Subjects	100
11.6.1	LOSO Cross Validation, 14 Channels, All Timbres . .	100
11.6.2	LOSO Cross Validation, 1 Channel, All Timbres . . .	101
11.6.3	LOSO Cross Validation, 14 Channels, Separate Timbres	103
11.7	Frequency Experiment - Separate Subjects	104
11.7.1	LOTiO Cross Validation, 14 Channels	104
11.7.2	LOTO Cross Validation, 14 Channels, All Timbres . .	105
11.7.3	LOTO Cross Validation, 8 Channels, All Timbres . . .	108
11.7.4	LOTO Cross Validation, 6 Channels, All Timbres . . .	112
11.8	Separate Subjects - Summary	115
11.9	Interpolation Results	115
11.9.1	Loudness Interpolation	116
11.9.2	Frequency Interpolation	117

12 Conclusion	118
13 Future Works	121

Part I

Overview

Chapter 1

Introduction

A Brain Computer Interface (BCI) is a communication system that enables the user to interact with a computer using his/her brain's electromagnetic signals. In the last decades, research in the BCI field has acquired an increased importance because low-cost devices have enabled BCI methodologies and theories to be applied in numerous fields of study. There are different technologies that can measure the brain's activity, these can vary in size, cost and capabilities. Depending on the field application of the BCI, special hardware and software are used. Some BCI technologies have an high *spatial resolution*, like the Magnetic Resonance Imaging (MRI). Those technologies are used to obtain detailed information about localization and amplitude of the brain signals. Technologies used for the analysis of the brain while performing an action or processing an external input need to have a higher *temporal resolution*, like electroencephalography (EEG). A higher *temporal resolution* allows the observation of changes in the state of the brain that happen within short periods of time.

In a BCI system that uses EEG technology, the electrical activity of the brain is read by a headset placed on the user's head. It has electrodes distributed symmetrically over the scalp and forehead. Initially the electrical potential, defined as the voltage difference between two electrodes originated by the neuronal activity of the brain, is detected [1]. Successively the EEG signals can be used to control a computer in real time or analyzed to find patterns or models of behavior.

The brain controls all the functions and activities of the body. For controlling movement, both conscious and unconscious, the brain sends electrical signals to the nervous system that in turn activate muscles or organs [2]. For interpreting the external world, changes in the environment trigger responses in the brain, and these can be attributed to input detection, recognition and processing. These two different functionalities can be categorized as *ac-*

tive or *passive*. *Active* behavior involves sending signals to the body, while *passive* behavior interprets the input from external sources. A BCI system can be tuned and modified to successfully detect one of these behaviors [3][4].

The use of a BCI constructed for *passive* behavior opens up the possibility to study human *emotional responses* from a new perspective. To be able to describe the emotion that a certain stimuli (visual, auditive or tactile) triggers on a person is a difficult task. The main reason is because emotions are subjective and the terminology used to describe them depends on the subjects and their cultural background. Nevertheless, evaluating changes in the state of the brain, without defining or describing emotion brings forth the possibility of measuring the *brain's response* objectively. Hence, the use of a BCI system, allows us to circumvent the limitations of subjective assessments by analyzing the brain's unbiased responses.

Therefore, this project aims to study the effect that audio stimuli induces on the brain. In particular, we study how variation of a single audio parameter affects the brain. The main objective is to model how the brain responds to controlled audio variations. Successively, further research could investigate what those brain variations represent in the emotional plane, potentially allowing to guide the user into a different state by varying the audio parameters. This is useful for various applications in fields spanning from the entertaining industry to music therapy.

The next section will define the aim of this project in a more rigorous manner, presenting the mathematical and statistical tools used to develop the cause-effect, audio to emotional state, understanding.

1.1 Project Statement

This project is a preliminary analysis on how single sound-parameter variations trigger different responses in the brain. The objective of this analysis is to find a direct relationship between a controlled single sound-parameter variations and the brain's signals by performing statistical regression analysis.

The relationship or model representing the brain-sound response is an objective and generalized measure of the brain's behavior to a controlled exposure of specific sounds. This model can then be used to infer what region or emotion gets triggered or heightened by the increase or decrease of a sound parameter. Note that this inference can now be based upon objective measures that are not dependent on a subject's perspective or the subject's

classification of their own emotional state. Furthermore, models for different sound parameters can be combined to speculate upon the expected response of modifying several sound parameters simultaneously.

Specifically, this project will focus on two different sound parameters, namely *loudness* and *frequency*. While these parameters will be varied to trigger some response, all remaining sound parameters will be fixed on a limited set of defined values.

This project will first seek to design an experiment that yields a representative amount of data for the regression analysis. This process begins with the selection of range and values of all variable and fixed parameters followed by the design and implementation of the experiment. The experiment is divided into two fragments, the *loudness experiment* and the *frequency experiment*. For the first experiment, *loudness* is the variable parameter while frequency, timbre and the sound envelope are fixed. For the second experiment, *frequency* is the variable parameter while loudness, timbre and the sound envelope are fixed. The objective of the experiments is to obtain the raw EEG signals of the listener while he/she is exposed to *loudness* or *frequency* variations. The EEG signals will be acquired using the Emotiv EPOC¹, a consumer grade EEG device.

The signals obtained in the experiments are pre-processed to remove noise, artifacts and to isolate certain frequency bands, that are associated with particular brain activities. This is accomplished by applying well known signal processing techniques that are standard in BCI. Furthermore, specific intervals of time, related to the brain's response, are selected as trials.

Using the covariance matrices of each trial, a measure of the distance between them can be obtained. Commonly in BCI, the distance between covariances are handled in Euclidean space, not taking into account that covariance matrices are symmetric positive definite (SPD) matrices and therefore belong to the curved space of a Riemannian manifold [5]. In this project we will consider the covariance matrices as points in the Riemannian space. These points we will be used to perform a statistical regression analysis in this space in order to estimate the brain's response signal as a function of single-sound parameter variation.

If a valid regression model is found, interpolation between two covariance matrices can be applied. The Riemannian interpolation can be used to map a given *loudness* or *frequency* value with its corresponding brain state, by approximating between two known covariance matrices.

¹<http://www.emotiv.com/>

1.2 Structure of the report

The report is sectioned into four parts:

Part I: Introduction

Chapter 1 contains a general introduction to the BCI field followed by the project statement. A summary of various related research and basic theories is described in Chapter 2. Chapter 3 gives a description of the EEG device used in the experiments and introduces the basic notation for the data obtained through the EEG device.

Part II: Theory

The notation and theory of the Riemannian manifold and its applications in the BCI field are explained in Chapter 4. It is followed by the descriptions of Regression and Interpolation theories in Chapter 5 and Chapter 6, respectively.

Part III: Sound Parameter Variation Analysis

Part III gives an overview of the BCI process constructed for this project. Chapter 7 elaborates on the design and construction of the loudness and frequency experiments. A detailed description of the pre-processing is given in Chapter 8. Data Analysis implementation is presented in Chapter 9. Finally, interpolation implementation is described in Chapter 10.

Part IV: Results and Conclusion

The results of the regression analyses for both experiments are shown in Chapter 11. The conclusion and the future works of the project are given in Chapter 12 and Chapter 13, respectively.

Chapter 2

Related Works

This chapter will provide an overview of the different topics that are used throughout the project. It serves as context for the experimental design and implementation and an introduction to BCI and emotions. It is divided into multiple subsections, since the project encompasses many different fields of study.

2.1 Emotion Classification

A classification of emotion or emotional reactions is a subjective and imprecise task. Being able to recognize emotion from audio stimuli is even more challenging, since the short-time descriptions of emotions are limited and could vary from person to person [6].

There are different emotion classification methods that propose solutions to the emotion recognition problem. One of them is the two dimensional emotional plane [6][7]. In this model, the valence level, varying from unpleasant to pleasant, represents the typology of the emotion, while the arousal level denotes the intensity of the emotion, varying from not aroused to excited. Another, more specific, model is the Thayer's emotion plane [8][6]. This model categorizes emotions into different classes, as shown in Figure 2.1, where these classes are organized into four quadrants that collectively represent the emotional plane.

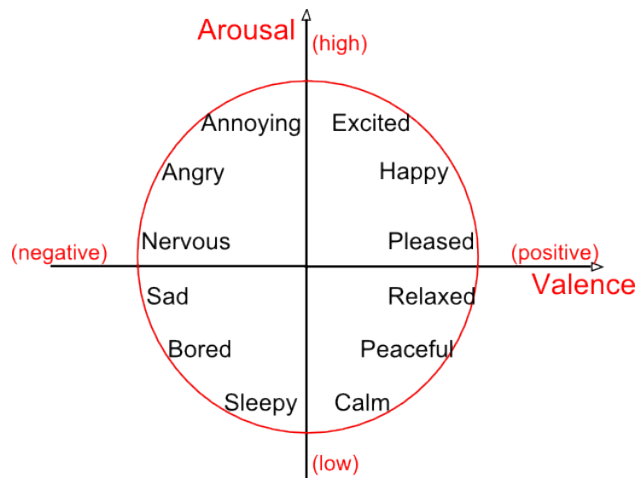


Figure 2.1: Thayer’s model of the Emotional Plane.

Many studies have been conducted to demonstrate that sound has the capacity to lead the user into a specific emotional state [9]. From music to non-musical sounds, different works argue that different contexts and musical/sound characteristics have an effect on the listener. Nonetheless, most research is based on the subject’s classification or description of their own emotion, making the results subjective and biased.

In the case of research based on non-subjective data, like BCI classifiers, it has been shown that some specific sounds, with specific characteristics, may have an impact upon the listener’s emotional response [10]. Although this is an improvement upon the subjectivity of other research, it presents a problem; it only applies to a specific combination of sound parameters and context and can not be extrapolated to other sounds. In other words, little research exists on the effects that sound parameter variations have on a listener’s emotional state.

2.2 Sound and Music Theory

The following subsections will give a review of the music and audio topics necessary to understand the audio parameters involved in the production and perception of sound. Note that the following sections follow the *Western common music notation system* [11].

2.2.1 Frequency and Pitch

Frequency is the measurement of the cycles of a repetitive waveform. This measurement is taken in Hertz (Hz) and it represents the number of cy-

cles per second. The human ear is limited to frequencies between 20Hz and 15,000Hz approximately [12] [11], although this range varies with age.

Pitch corresponds to the perceptual auditory attribute of a frequency, a subjective measure that allows sounds to be ordered on a scale from low to high. It is limited to the human frequency-range of hearing and affected by loudness, sound envelope and the presence of other simultaneous frequencies [11].

The human ear and brain process the frequency or pitch of a sound together. Some theories explain that each frequency has a corresponding displacement in a membrane, called Basilar membrane, located inside the inner ear [13], while other theories attribute pitch recognition exclusively to processing in the brain. Both, Basilar membrane displacement and brain processing occur as a response to listening to a pitch. Nevertheless, classification of the pitch is attributed to the brain [11].

2.2.2 Timbre, Tone and Note

Timbre, also addressed as *sound quality*, corresponds to the properties of sound that allow humans to distinguish between different sources of sound. It is the collection of sinusoidal components produced by a source or instrument, and it consists of several frequencies with different strengths [13]. It also corresponds to the dynamic characteristics of the sound, like vibrato or the sound envelope, further explained in Subsection 2.2.4.

As shown in Figure 2.2, the individual sinusoids that comprise a pitch produced by a source are called *partials* or *components*. The first component is the fundamental frequency. Other frequencies and their corresponding amplitude, along with the sound's dynamics, make a timbre characteristic, therefore allowing a human to classify it as a specific instrument or any other known source [11].

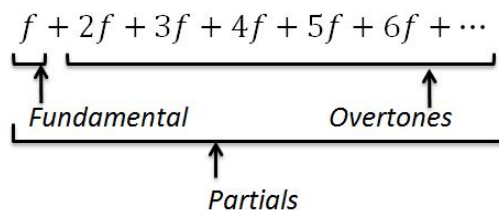


Figure 2.2: Harmonic Overtone Series.

Tone is characterized by three qualities, pitch, musical loudness and timbre. Then the addition of temporal qualities, like onset and notation create a *note* [11]. Tone, because it includes timbre, can also be understood as the particular sound of a specific instrument or sound source with the addition of a defined volume and pitch. This encompasses all parameters or components of the sound that make it differentiable or distinguishable from other sounds [13].

2.2.3 Fundamental Frequency and Harmonics

The lowest of all the frequency *components* in a tone corresponds to the *fundamental frequency* and it usually is the predominant, most discernible sound in the combination of *partials*. All the other components above the *fundamental frequency* are also called *overtones*.

Overtones can be classified as *harmonic* or *inharmonic*. If the *overtones* are positive integer multiples of the fundamental then they are *harmonic*. Otherwise, *overtones* are considered *inharmonic*. *Harmonic* sources are normally melodic or harmonic instruments and carry tunes and notes throughout songs. Figure 2.3 shows different signals over time and their corresponding overtones. Note that the representation of a signal in the frequency domain is defined as spectrum. The amplitude of a wave is the distance from its peak height to zero, while the magnitude is defined as the absolute value of the difference between the positive and negative peak.

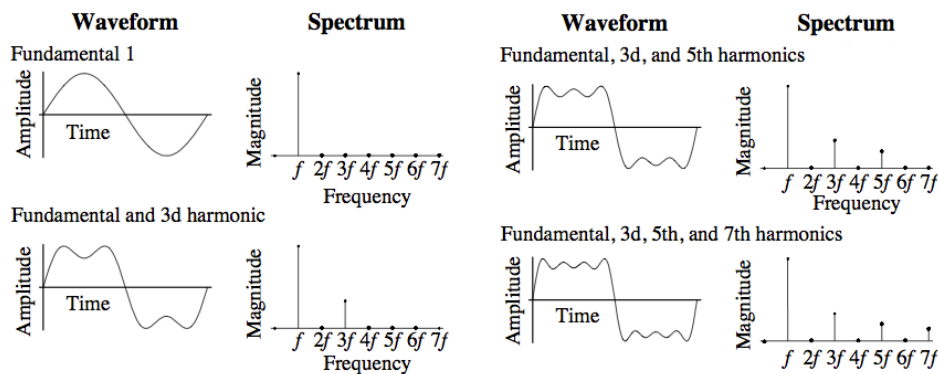


Figure 2.3: *Harmonics Waveform and Spectra* [11].

Inharmonic sources are normally percussions, drums, and other noises containing overtones that are non-integer multiples of the *fundamental frequency* and therefore not agreeable to *diatonic* scales and pitches. An example of inharmonic source is shown in Figure 2.4

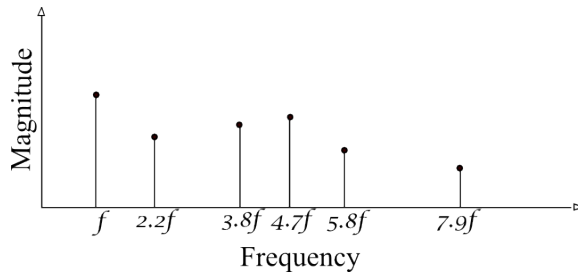


Figure 2.4: Example of an inharmonic spectrum.

A spectral analysis of all the components of a tone over small periods of time is shown in Figure 2.5. In this Figure it is possible to observe the relationship between *fundamental frequency*, *overtones* and their corresponding strength and decay overtime.

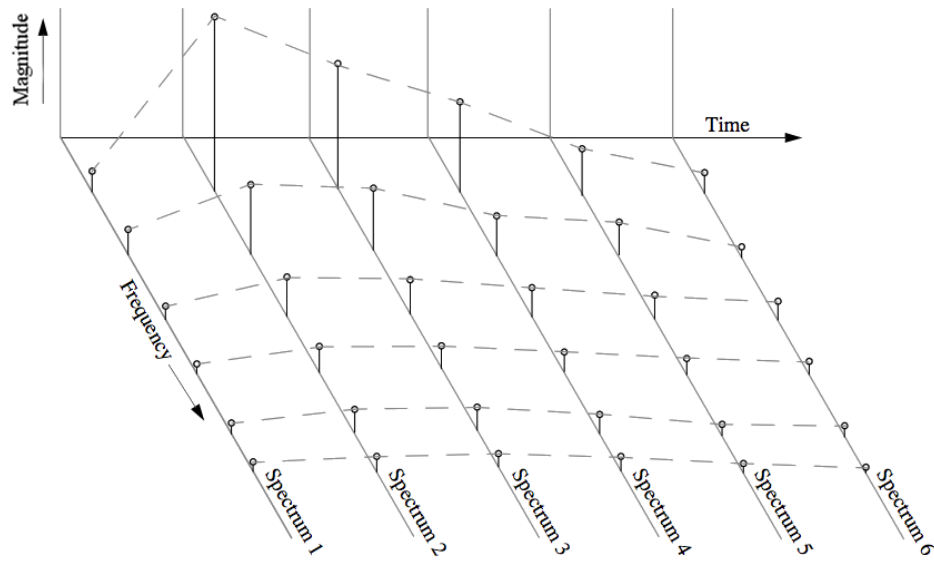


Figure 2.5: Dynamic spectrum. The amplitude variation over time is shown by the dashed lines that connect the partials for each frequency [11].

2.2.4 ADSR Envelope

As stated before, a tone can be represented as the sum of all its *partials* over time. As shown is Figure 2.5, each partial has a different, changing amplitude over time. The average amplitude of all the partials over time is called an *amplitude envelope* [11] or attack-decay-sustain-release (ADSR)

envelope. Figure 2.6 shows the amplitude envelope of the sound represented in Figure 2.5.

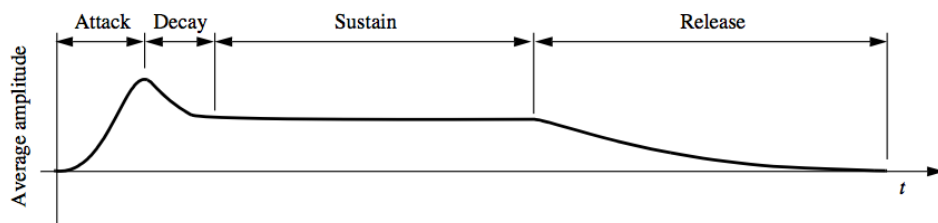


Figure 2.6: Amplitude envelope of waveform [11].

The amplitude envelope can be divided in four segments. The *attack* is the time between the onset or first energy release of a source until the source is at maximum dissipation of energy. The *decay* is the time between the maximum energy and the decline to a sustained level of energy. The *sustain* is the time the sound is maintained as a stable dissipating energy that matches the exciting force of the source. The last fraction of the envelope is the *release* and it represents the time from the moment no more sound is produced until all the energy has dissipated and there is silence [11].

2.2.5 Loudness

Loudness is a measurement of the intensity of a sound. The intensity of a sound is the energy with which the sound waves flow through the eardrums and into the inner ear [11]. Because human ears are sensitive to pressure, and because pressure is easier to measure, it is used as a measure of loudness [13].

Sound pressure is directly related to the amplitude of the sound wave, and it has a wide range of variation. The sound pressure for sound sources can vary from $20\mu Pa$ to $20Pa$. Thus sounds can vary over a range of amplitudes which is greater than a million. Because of this, and the way we perceive sound, the sound pressure level (SPL) is usually expressed in dB, on a logarithmic scale [13] [11]. The scale uses the actual pressure level in Pa, and a reference pressure level of $20\mu Pa$ at $1kHz$ [13] [14] [11].

$$dB(SPL) = 20 \log\left(\frac{P_{actual}}{P_{ref}}\right) \quad (2.1)$$

The range of values between 0dB and 120dB is easier and clearer for expressing variations in loudness that can be perceived by the human ear, without pain or damage for the auditive system [13]. This scale uses a small set of values that describe a wide range of intensities [11]. For example, the

useful range of musical intensities is between 45 dB to 95 dB considering the wide range of timbres and frequency ranges of all instruments [11]. Using this scale, Figure 2.7 shows the threshold of human hearing and the upper limit, when loudness is no longer perceived and replaced by pain.

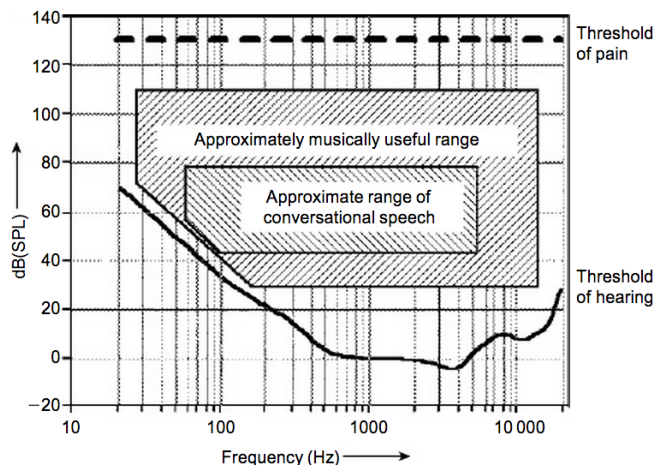


Figure 2.7: The average threshold of hearing and threshold of pain with approximate musical and speech ranges [13].

Differently from the physical world, where the analog sounds are measured using the dB scale, digital sounds are measured using a different scale, referred as dB Full Scale or dBFS. It is usually measured down from the limit of the loudest sound that can be reproduced with no distortion. This scale still measures loudness as pressure but it is relative to the hardware that reproduces and records it. A reference loudness is selected and assigned a dB value of 0dBFS [11], this reference is the highest possible loudness in the system. Any other sound in the system is measured compared to the reference, thus all sounds are assigned either 0 or a negative dBFS value.

2.2.6 Perception of Sounds

Amongst all the parameters that define different sounds, there are some *objective* parameters that are measured in the world outside the listener and other *subjective* parameters that depend on the perception and experience of the listener. An objective measurement of the subjective perception of the listener is hard to obtain, not only because of the subjectivity of the listener's descriptions, but also because of the nonlinear nature of sound parameters.

Pitch and loudness for example, are not linearly proportional to frequency and intensity [11]. The nonlinear nature of these parameters requires a transformation between their real-world scale to scale representative of human perception. Besides nonlinearity, sound parameters are all related to

each other in a *nonorthogonal* manner [11]. Therefore the influence between parameters is a complex relationship that is not to be considered lightly in the measurements of sound perception.

Loudness Perception

Perception of loudness is not as straightforward as perception of pitch. While intensity is the most important factor affecting loudness, it is also affected by other factors, especially frequency [11]. The ear is a pressure sensitive organ that divides the audio spectrum into a set of overlapping frequency bands whose bandwidth increases with frequency [13]. In other words, the pressure amplitude of a sound wave does not only relate to its perceived loudness, our hearing capabilities are different for different frequencies. In Figure 2.8 the curves show how loud a sound has to be (dB), to be perceived to have the same loudness as a base reference.

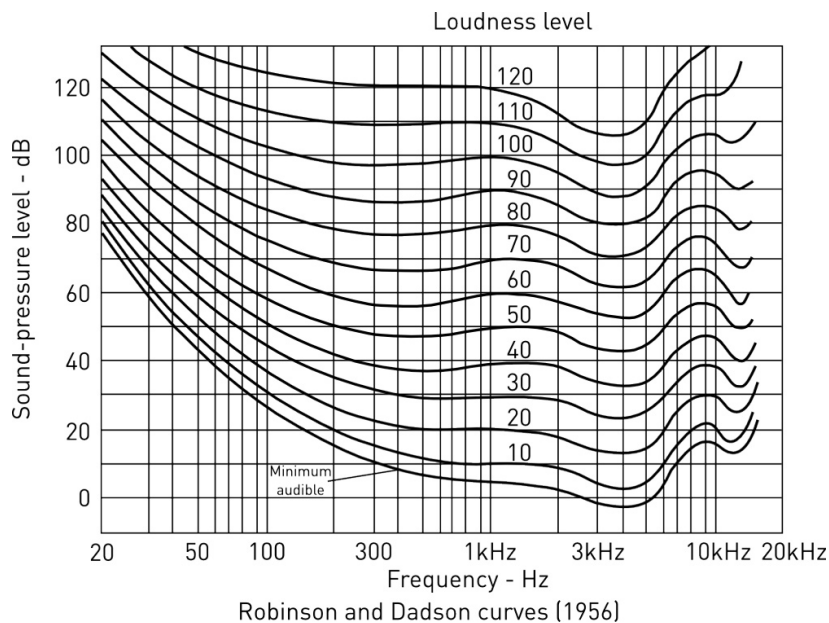


Figure 2.8: Loudness contours for the human ear [13].

A scale used to measure loudness in terms of both frequency and amplitude is the *phon* scale [13]. The loudness level in *phon* of a *sound* is defined as the sound pressure level (SPL) in decibels (dB) of a standard 1kHz pure tone when the *sound* and the standard tone are equally loud [14]. The *phon* scale identifies equal loudnesses across all perceivable frequencies and intensities. As can be observed in Figure 2.8, sounds with a low frequency need to have a higher intensity (or dB level) to be perceived as clearly as sounds

with a frequency close to $1kHz$. Note that the change of sensitivity of the human ear above $1kHz$, shown in the curves in Figure 2.8, is due to the construction of the external ear, which amplifies certain frequencies better.

2.2.7 Diatonic and Chromatic Scale

The diatonic scale is commonly used as the prototype for all other scales. It contains seven pitches per octave named: C, D, E, F, G, A, B. It has two different interval sizes a *whole tone* and a *semitone*, a whole tone is equal to two semitones. The chromatic scale extends the diatonic scale by breaking the whole tones into semitones, producing a scale with 12 pitches per octave: C, (C \sharp or D \flat), D, (D \sharp or E \flat) E, F, (F \sharp or G \flat), G, (G \sharp or A \flat), A, (A \sharp or B \flat), B [11].

To further specify the tuning of each pitch in the scale, the equal-tempered scale specifies all the frequencies using the same uniform semitone interval. Choosing A440 as a frequency reference, A in the 4th octave of a piano, all the intervals in the octave are calculated. Further octaves are calculated using that octave as a reference. Equation 2.2 shows how to calculate each frequency using the octave k and interval v [11].

$$f_{k,v} = f_R \cdot 2^{v+k/12} \quad (2.2)$$

Using equation 2.2 it is possible to calculate all notes in all octaves without inconsistencies. By using this system, music based on different scales and instruments can be played with the same tuning [11].

2.3 Emotional Response to Sound Parameters

2.3.1 Loudness

There is a wide variety of studies concerning the different effects of loudness. It has been shown that changing pressure level or loudness of a sound can directly affect both emotion valence and intensity [15] [16]. Other studies also found that an increase in intensity is also perceived as a warning cue [17].

Furthermore, tests have shown that the perception of loudness is influenced by the emotional state of the subject [18]. Some states cause the subject's auditory stimuli to be perceived as louder and fear-inducing. For example, when sound stimuli are paired with a hostile environment or when a frightening memory is recalled [19].

In works more closely related to the objective measurement of sound perception, using a Functional Magnetic Resonance Imaging (fMRI), experiments show that the loudness level of a stimuli is related to the cortical activation of the brain [20].

2.3.2 Frequency

Studies concerning frequency and pitch can be found across many different fields of studies. This is due to the diverse applications and strong effects that pitch have on humans. For example, pitch is a discriminant factor in the design of warning sounds for dangerous situations. These studies show that a high pitch sound suggests more urgency than a low pitch sound [21][22]. Other studies show that an increase in sharpness (combination of higher frequencies and pure tones) provokes major discomfort and negative effects on the listener [9][15].

In the context of horror games, it has been found that there is an instinctive fear reaction to certain frequencies. Frequencies within a very low range or very high range produce a fear response. The low range frequencies are associated with predators while high range frequencies are associated with human screams [9][23].

2.3.3 ADSR

The ADSR envelope has also been found to affect the emotional response of the listener [24]. Previous research has shown that different sound energy events are associated with different affects [24]. For example, in a video game context, it was demonstrated that an audio signal with fast attack and release give a perception of urgency [25] while long attack periods can be used to create tension [23].

2.3.4 Sound Source And Localization

In the past years it has been argued that a central aspect of human interaction with sound is the ability to perceive the position of a sound source [26] and the surrounding auditory space[27]. For example, sounds positioned behind the listeners are more arousing than sources in front of the listeners and this effect is more pronounced for natural sound sources, compared to artificial sources.

In the context of horror games intentional ambiguity of a sound's source and location builds suspense and terror in the player [28]. Beyond the specific position of the sound, variation in the dimension of the space surrounding

the listener can also modulate the emotional response.

2.3.5 Sound Effects

Sound effects are artificial processes that modify certain properties of a sound. A sound effect could modify any of the sound parameters explained above and create a very clear reaction or emotional response. For example, effects like reverberation or delay can reproduce the same emotional response as being placed in a large environment [26].

It has been demonstrated that applying an effect like low-pass filter to a musical sample, produces a relaxed state. In contrast the use of a high-pass filter effect produces a more agitated state [29].

2.3.6 Signal to Noise Ratio (SNR)

In order for a sound to be heard it needs to be greater in strength than the ambient noise. A measurement of this criteria is the signal to noise ratio (SNR).

$$SNR = \frac{\textit{Signal intensity level}}{\textit{Ambient noise intensity level}}$$

The SNR is another sound parameter that can determine variation in the listener emotional state. For example low sound quality or low SNR may rise an uncertain sensation due to the difficulty in identifying and localizing the signal sources [30] [25].

2.4 Emotional Response to Music

It is important to distinguish between the emotional response to music and the emotional response to sound. Although they are related and similar in many ways, the context is different and therefore the emotions and reactions of the subjects are different.

The musical features, parameters and properties presented below correspond to musical fragments, musical instruments and voices. Therefore, some of these features are comparable to sound parameters while others are exclusively musical. Nevertheless, research of the emotional response to music enables a more general understanding of the effect *sounds* have on the human brain. Some of the findings presented below are helpful to understand the expected emotional reaction to different sounds parameters. Note that

in the following subsections some of the characteristics directly related to the loudness and frequency sound parameters, investigated in this project, are *highlighted*.

2.4.1 Musical Features

In the field of music theory, it is possible to find research that determines which musical structures create the emotional expressions in a western musical piece or score [31] [32] [33]. For example, it has been found that tempo, mode, *pitch level*, harmony and rhythm are the features that affect a listener more intensely [31].

Happiness vs Sadness

Juslin encapsulates the musical features that are more discriminating in the expression of sadness and happiness [34].

Sadness features: Slow mean tempo, several tempo variations, *low sound level*, legato articulation, slow vibrato, small-uniform articulation variability, slow tone attacks, flat micro-intonation and end-of-phrase *ritardando*.

Happiness features: Fast mean tempo, few tempo variations, *fairly high sound level*, *staccato articulation*, contrasting articulation variability, little sound level variability, *bright timbre*, fast tone attacks and rising micro-intonation.

Classification of Five Emotional Targets

The relationship between emotions and musical-audio features has been tested by Laurier [35]. In this experiment, 110 film soundtrack excerpts, fifteen seconds each, were evaluated by the participants using 5 emotional targets (fear, anger, happiness, sadness, tenderness) and a three dimensional model (valence, energy arousal and tension arousal) [35]. Different types of audio features based on temporal and spectral representation of the audio signal were extracted:

- **Timbral:** Barkbands, Mel Frequency Cepstral Coefficients, pitch valence, loudness.
- **Spectral:** Flatness, flux, complexity, skewness, crest, decrease, speed.
- **Tonal:** Dissonance, chords change rate, mode, key strength, tuning, diatonic strength.

- **Rhythmic:** Bpm, zero-crossing rate, silence rate, onset rate, danceability.

From the evaluation of all the different audio features, several showed a high correlation with different emotional states.

- Dissonant sounds have an high correlation with anger.
- Major mode has a high correlation with happiness and tenderness, while minor modes have an high correlation with sadness and fear.
- It confirms the hypothesis that faster tempo is correlated with happiness [34].
- *Loudness* is more highly correlated with happiness than anger.

Expression of Single Tones

A similar study, based on an opera singer's voice, explores the features that contribute mostly to the emotional expression of single tones. Note that most of the features also applicable to wind instruments [36]:

- Onset of phonation (voicing).
- Vibrato.
- *Excitation of higher harmonic partials.*
- Gradual pitch increase from the onset to the sustained stage (transition).
- An abrupt pitch increase at the very onset of the tone.
- Pitch change within the tone.
- Unit pulse (a feature produced by the vocal cords).

2.5 Brain and Emotion

Several studies have shown that there is a lateralization in the brain's activation, related to emotional response [37]. Recent studies demonstrate that positive emotions, like happiness, produce a greater EEG activity on the left hemisphere. In contrast to negative emotion, like anger, that yields a greater

EEG activity in the right hemisphere [38][39][40].

Furthermore, there are also some studies conducted with musical samples that support this theory. It was found that pleasant music produces greater left-frontal activity in the listener while unpleasant music produces greater right-frontal activity [41]. Other studies, using music samples, have been conducted on the theta band (4-8 Hz), the studies show that there is ample variation in the frontal mid-line (Fm) theta power. An increase of power is associated with pleasant music while a decrease of power is associated with unpleasant music [42] [38].

Chapter 3

EEG Technology

In a BCI system using EEG technology, the signals are read by a series of electrodes, normally placed on a headset. The data collected by the headset can be analyzed to find how the brain reacts to external stimuli. The amount of information that can be obtained from the headset depends on the quality and resolution of the headset's hardware.

3.1 Emotiv EPOC

This project uses the consumer grade EEG device, Emotiv EPOC shown in Figure 3.1.



Figure 3.1: The Emotiv EPOC Headset

The Emotiv EPOC has a sample resolution of 128 Hz and it is composed of 14 recording electrodes and 2 reference electrodes. The voltage difference between each of the 14 recording electrodes and their corresponding reference electrode defines a channel. Besides the electrodes, the Emotiv EPOC

headset has a two axes gyroscope that can be used to detect head movements.

The Emotiv EPOC sends the electrode's readings to a computer through a wireless Bluetooth USB port. The headset comes with a proprietary software called TestBench. Through the TestBench, showed in Figure 3.2, it is possible to verify the contact quality of each of the 14 measuring electrodes and the 2 reference electrodes. The same program records the EEG signals and exports them to European Data Format (EDF) or Comma Separated Values (CSV) files.

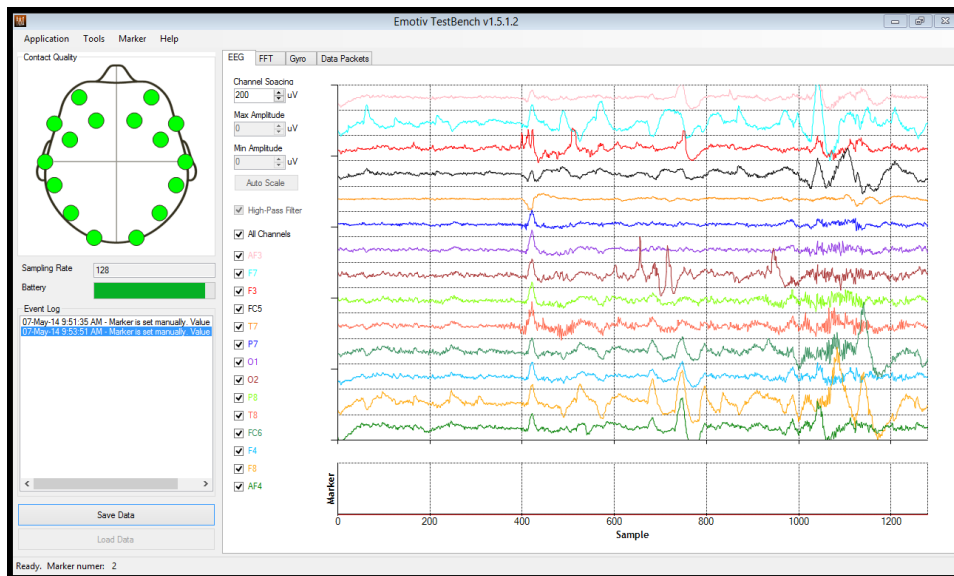


Figure 3.2: TestBench

3.2 Electrode Placement

Conventionally, the human brain is divided into six main areas called lobes [3]. It is possible to visualize four of these lobes, as seen in Figure 3.3, from a lateral point of view of the cerebral cortex. The limbic lobe and the insular cortex are located on the interior part of the brain and therefore not shown in Figure 3.3.

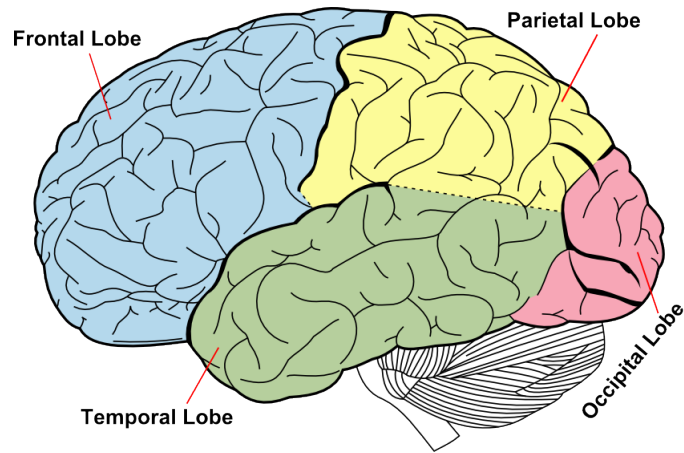


Figure 3.3: A side view of the human brain. The Frontal lobe is located above the the eyes and nose, the Temporal lobe is located around the ears and the Occipital lobe is located on the back of the cranium.

The electrodes of a EEG device are mapped on the human head using the different lobes as reference. Normally the electrodes are identified by using one or two letters corresponding to the nearest brain lobe(s) or region, as seen in Table 3.1 [3].

Electrode	Brain Area
Fp	Frontal polar site
F	Frontal
AF	Antero-Frontal
T	Temporal
C	<i>Central</i>
P	Parietal
O	Occipital

Table 3.1

Note that the letter C is only used as a reference. The region letters are then followed by a number or a lowercase letter z [3]. Even numbers are used for the electrodes placed on the right side of the scalp, while odd numbers are used for the left. The letter z identifies electrodes placed on the mid-line of the scalp. The higher the number the farther the electrode is located from the mid-line. For example FC5 is located on the left side of the brain between the Frontal lobe and the central area.

A commonly used electrode placement convention is the 10-10 System [43], this system follows the same notation explained before. This system, showed in Figure 3.4, derives from the 10-20 System and adds electrodes to increase its spatial resolution [44].

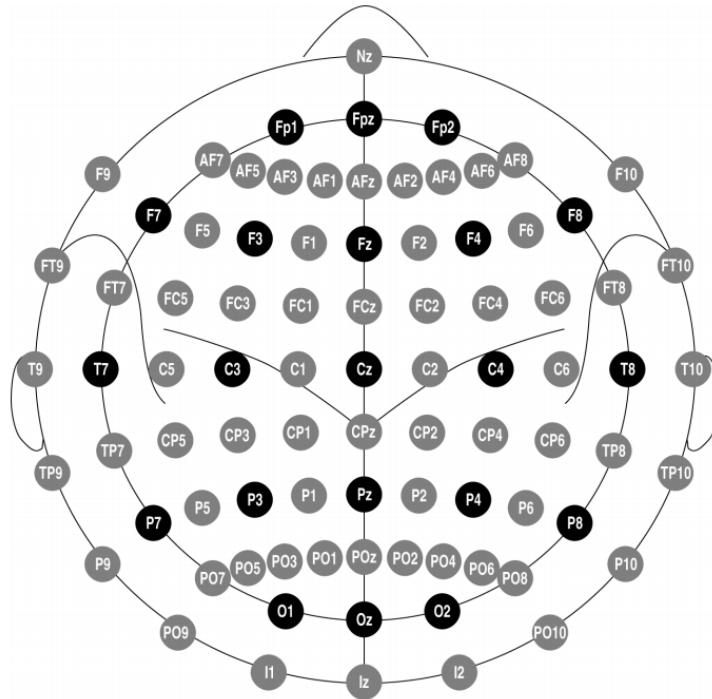


Figure 3.4: The 10-10 system electrodes in gray and black, the black circles represent the electrodes of the 10-20 System [43]. The Nz , Fpz , Oz and Iz circles are used as references for placing the other electrodes. The line from Nz to Iz is the mid-line.

The 14 electrodes placed on the Emotiv EPOC headset correspond to locations in the 10-10 System. These electrodes are **AF3**, **AF4**, **F7**, **F8**, **F3**, **F4**, **FC5**, **FC6**, **T7**, **T8**, **P7**, **P8**, **O1** and **O2** as shown in Figure 3.5.

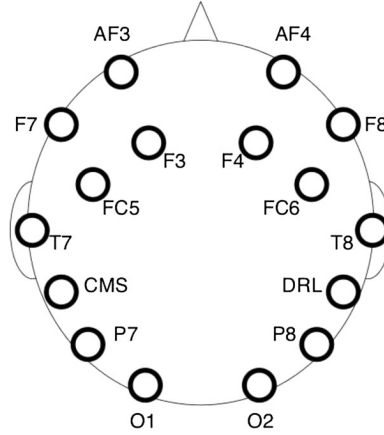


Figure 3.5: Emotiv EPOC electrodes, based on the 10-10 System [45]. The electrodes denoted as CMS and DRL represent the two reference electrodes P_3 and P_4 respectively.

3.2.1 Brain Rhythms

The EEG signals obtained from the scalp are composed of several signals at different frequency bands. These different frequency bands are called rhythms. The sum of all frequency bands is referred as rhythmic activity. Each of these rhythms can be associated with different activities. Table 3.2 gives an overview of these rhythms and activities. Variations in a specific rhythm indicates activity in its corresponding brain function [3].

Rhythm	Frequency	Location	Associated Activities
Delta (δ)	<4 Hz [46]	Thalamus	Sleep/Unconsciousness, Concentration [47]
Theta (θ) [48]	4–7 Hz	Neocortex	Sleep and spatial navigation [49]
Alpha (α) [50, 46]	8–14 Hz	Occipital lobe	Resting and Relax
Mu (μ) [51]	8–13 Hz	Sensorimotor Cortex	Motor Imagery (MI)
Beta (β)	14–30 Hz [52]	Sensorimotor Cortex	MI, active thinking, high alert

Table 3.2: Brain rhythms, location and some of their associated activities [3].

3.3 Notation

The following section introduces the basic notation used to describe and manipulate the information obtained from the EEG headset.

3.3.1 Channels, Samples and Trials

A complete data set contains all the information obtained from one or more experiments and any number of subjects. It is comprised of all the samples over time of all the channels.

A channel represents the electric potential measured between an electrode and its reference electrode [5] [3]. The set of all channels is defined as $\mathcal{C}_{ALL} = \{c_1, c_2, \dots, c_N\}$, where N is the total number of channels used to construct the data set. Every sample corresponds to a time point $t \in \mathcal{T}_{ALL}$, where $\mathcal{T}_{ALL} = \{t_1, t_2, \dots, t_T\}$ is the set of all sampled time points T .

A data set can be divided into a set of I trials. Each trial i , $1 \leq i \leq I$ contains a subset of samples $\mathcal{T} \in \mathcal{T}_{ALL}$ that corresponds to a specific time interval. Furthermore, a trial i may contain the information of all channels \mathcal{C}_{ALL} or a subset of channels $\mathcal{C} \in \mathcal{C}_{ALL}$.

Let us define the row vector containing a set of samples \mathcal{T} for one channel as $\mathbf{x}_n(\mathcal{T})$, where n is the channel number and $1 \leq n \leq N$. A trial i using a set of N channels is represented in the form of a matrix \mathbf{X}_i :

$$\mathbf{X}_i = \begin{bmatrix} \mathbf{x}_1(t_1) & \cdots & \mathbf{x}_1(t_S) \\ \vdots & \ddots & \vdots \\ \mathbf{x}_N(t_1) & \cdots & \mathbf{x}_N(t_S) \end{bmatrix} \quad (3.1)$$

where t_1 is the first sample in the time interval \mathcal{T} and S is the number of samples in trial i .

3.3.2 Spatial Covariance Matrix and Sample Covariance Matrix (SCM)

Let us define $\mathbf{x}_{\mathcal{C}}(t)$ as the column vector containing the samples at time t for the set of channels $\mathcal{C} \subseteq \mathcal{C}_{ALL}$ and $\mathbf{x}_{\mathcal{C}_{ALL}}(t) \in \mathbb{R}^N$ as the column vector containing the samples at time t of all N channels. The *Spatial Covariance Matrix* is defined by [5]:

$$\Sigma = E \left\{ (\mathbf{x}_{\mathcal{C}_{ALL}}(t) - E \{ \mathbf{x}_{\mathcal{C}_{ALL}}(t) \}) (\mathbf{x}_{\mathcal{C}_{ALL}}(t) - E \{ \mathbf{x}_{\mathcal{C}_{ALL}}(t) \})^T \right\} \quad (3.2)$$

where T denotes matrix transposition and $E\{\cdot\}$ denotes the expected value over time. An unbiased estimator for the covariance matrix Σ is the *Sample Covariance Matrix* (SCM).

The SCM for trial i , $P_i \in \mathbb{R}^{N \times N}$ is defined as [5]:

$$P_i = \frac{1}{S-1} \mathbf{X}_i \mathbf{X}_i^T \quad (3.3)$$

Note that the SCM matrix P_i can be also defined for a subset of n channels, $P_i \in \mathbb{R}^{n \times n}$, with $1 \leq n \leq N$.

Part II
Theory

Chapter 4

Riemannian Geometry

At the end of the XIX century, Georg Riemann presented a new type of geometry that allowed the description of non-Euclidean geometric structures. Euclidean geometry sets the foundations for studying a Vector space. As shown in Figure 4.1, in this space it is possible to draw a straight line between any pair of points. This line is unique and can be infinitely long in both directions. The internal segment between these two points is the shortest possible curve between the two points [53] [54].

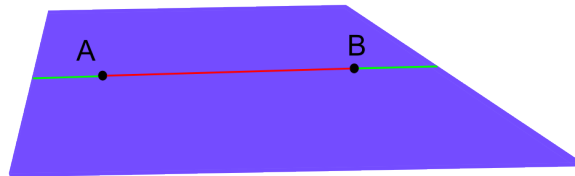


Figure 4.1: Two points, A and B , represented on Euclidean space. The line that connects A and B is the shortest distance between them, a straight line.

In contrast with Euclidean geometry, Riemannian geometry is the subset of geometry that studies curved spaces. In a Riemannian space, the supposition is that the space has a positive curvature, like a sphere or cylinder [55]. An example of a space with a positive curvature is shown in Figure 4.2. In this space, the shortest line that connects two points is no longer a straight line but a curve. This curve, which connects two points locally, is called geodesic and its length represents the distance between these points.

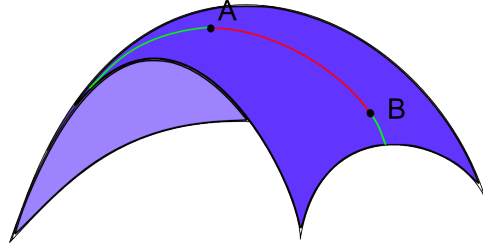


Figure 4.2: Two point, A and B , on a Riemannian space. The minimum curve that connects these two points is called geodesic.

The next section will give an overview of the mathematical operations in a Vector space used for the project. Section 4.2 presents the framework of the Riemannian manifold and Riemannian distance is explained in Section 4.3. The last part of this chapter is centered on the computation of the geometric mean on the Riemannian manifold.

4.1 Operations on Vector Space

The following sections present some valid mathematical operations in a Vector space that will be useful for intermediate computations in the more restricted Riemannian space.

4.1.1 Inner Product on a Vector Space

Let V be a Vector space. An inner product on V is a mapping $g : V \times V \rightarrow \mathbb{R}$ that has the following properties [56]:

Linearity

$$g(u_1 + u_2, v) = g(u_1, v) + g(u_2, v), \forall v, u_1, u_2 \in V$$

$$g(\alpha u, v) = \alpha g(u, v), \forall u, v \in V \text{ and } \forall \alpha \in \mathbb{R}$$

Symmetry

$$g(u, v) = g(v, u), \forall u, v \in V$$

Positiveness

$$g(u, u) \geq 0, \forall u \in V \text{ with } g(u, u) = 0 \Leftrightarrow u = 0$$

4.1.2 Frobenius Norm

Let M be the Vector space of the $m \times n$ matrices. The Frobenius norm (or vector norm) is a matrix norm of a matrix $P \in M$, defined as [57] [58]:

$$\|P\|_{\mathbf{F}} = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |p_{ij}|^2} \quad (4.1)$$

where p_{ij} are the elements of the P matrix, with $1 \leq i \leq m$ and $1 \leq j \leq n$.

This definition is equivalent to [5] [57]:

$$\|\mathbf{P}\|_{\mathbf{F}} = \sqrt{\text{Tr}(\mathbf{P}\mathbf{P}^T)}, \quad (4.2)$$

where $\text{Tr}(\mathbf{P}\mathbf{P}^T)$ represents the matrix trace of $\mathbf{P}\mathbf{P}^T$ and \mathbf{P}^T is the conjugate transpose [57].

4.1.3 Vectorizing a Matrix

As explained in Section 1.1, the aim of this project is to find a relationship between a single sound-parameter variations and the brain's response using a regression model. For this purpose, the vectorization of a matrix is needed to transform covariance matrices into features for the regression model. Further description of regression methods will be presented in Chapter 5.

Given a $n \times n$ symmetric matrix P, it is possible to represent it as a column vector of size $\frac{n(n+1)}{2}$ by vectorizing the upper triangular section of the matrix, as shown in Equation 4.3 [59].

$$\text{vect}(\mathbf{P}) = \mathbf{p} = \left[p_{(1,1)}, \sqrt{2}p_{(1,2)}, \dots, \sqrt{2}p_{(1,n)}, p_{2,2}, \sqrt{2}p_{(2,3)}, \dots, \sqrt{2}p_{(2,n)}, \dots, p_{(n,n)} \right]^T \quad (4.3)$$

Here, $p_{(i,j)}$ represents the (i, j) th elements of the matrix P and the term $\sqrt{2}$ is applied to all non-diagonal elements of P to preserve the norms relation: $\|\mathbf{P}\|_{\mathbf{F}} = \|\text{vect}(\mathbf{P})\|_2$ [59]. The term $\|\cdot\|_2$ denotes the \mathcal{L}_2 norm of a vector [5] [60].

Equally, it is possible to apply the inverse operation $\text{unvect}(\mathbf{p})$ to obtain the matrix P without loss of information [59]:

$$\text{unvect}(\mathbf{p}) = \mathbf{P} = \begin{bmatrix} p_{(1,1)} & p_{(1,2)} & \cdots & p_{(1,n)} \\ p_{(1,2)} & p_{(2,2)} & \cdots & p_{(2,n)} \\ \vdots & \cdots & \ddots & \vdots \\ p_{(1,n)} & p_{(2,n)} & \cdots & p_{(n,n)} \end{bmatrix} \quad (4.4)$$

The following sections address the main theory behind the Riemannian manifold and its practical applications in BCI.

4.2 Differentiable Manifold

By definition, a differentiable curve can be locally approximated to a straight line. The same logic applies to an n -dimensional differentiable manifold \mathcal{M} . This manifold can be locally approximated to an n -dimensional Euclidean space. For differentiable manifolds, it is possible to define the derivatives at a point P . These derivatives lay on the vector space T_P , which is the tangent space at point P of the manifold \mathcal{M} .

4.2.1 Riemannian Metric

A Riemannian metric g on a differentiable manifold \mathcal{M} is a function that defines for every point P in \mathcal{M} , an inner product g_P in the tangent space T_P [56]. Thus, a Riemannian metric on \mathcal{M} is a family of inner products $g_P : T_P \times T_P \rightarrow \mathbb{R}, \forall P \in \mathcal{M}$. Using this metric it is possible to define the length of a curve (or geodesic) between two points, which represents the Riemannian distance between these two points, as described in Section 4.3.

4.2.2 Riemannian Manifold

A Riemannian manifold is a pair (\mathcal{M}, g) where \mathcal{M} is a differentiable manifold and g is a Riemannian metric on \mathcal{M} [56].

4.2.3 Symmetric Positive Defined Matrices

Let us define $M(n)$ as the space of $n \times n$ real matrices, $\mathcal{S}(n)$ as the space of all symmetric matrices in $M(n)$, and let $\mathcal{P}(n)$ be the set of all $n \times n$ symmetric positive-definite (SPD) matrices [61] [5]. Formally,

$$\mathcal{S}(n) = \{S \in M(n), S^T = S\} \quad (4.5)$$

$$\mathcal{P}(n) = \{P \in \mathcal{S}(n), \mathbf{a}^T P \mathbf{a} > 0, \forall \mathbf{a} \in \mathbb{R}^n\} \quad (4.6)$$

Note that the *Sample Covariance Matrices* (SCM) $\in \mathcal{P}(n)$.

4.2.4 Logarithm and Exponential of a Matrix

Given a matrix $P \in \mathcal{P}(n)$, a matrix U of P 's eigenvectors and the eigenvalues σ_i of P , where $0 < \sigma_1 < \sigma_2 < \dots < \sigma_n < \infty$ and $\sigma_n > 0$. The exponential matrix of P can be computed as [5]:

$$\exp(P) = U \text{diag}(\exp(\sigma_1), \dots, \exp(\sigma_n)) U^T \quad (4.7)$$

Similarly, the inverse operation is computed using the logarithm, as shown in Equation 4.8:

$$\log(\mathbf{P}) = \mathbf{U} \operatorname{diag}(\log(\sigma_1), \dots, \log(\sigma_n)) \mathbf{U}^T \quad (4.8)$$

The following section will present a method to compute the distance between two points on the Riemannian manifold. Successively, after defining this distance, it is possible to compute the mean between a set of points that lay on a manifold.

4.3 Riemannian Distance

In the space of SPD matrices, where $\mathcal{P}(n)$ is a differentiable Riemannian manifold \mathcal{M} , every matrix \mathbf{P} can be represented as a point in \mathcal{M} . For each point $\mathbf{P} \in \mathcal{P}(n)$, there is a set of tangent vectors $\mathbf{S} \in \mathcal{S}(n)$ defined as the derivatives of \mathbf{P} on \mathcal{M} [5]. The tangent space $T_{\mathbf{P}}$ at point \mathbf{P} is comprised of $\mathcal{P}(n)$, the set of all tangent vectors \mathbf{S} [61]. Note that an element of the tangent space is usually denoted as a tangent vector even if \mathbf{S} represents symmetric matrices [62].

It is possible to define, for each tangent space, a local inner product $\langle \cdot, \cdot \rangle_{\mathbf{P}}$. The *natural metric* on the manifold of SPD matrices is defined by the positive local inner product, shown in Equation 4.9, and its induced norm, shown in Equation 4.10 [5]:

$$\langle \mathbf{S}_i, \mathbf{S}_j \rangle_{\mathbf{P}} = \operatorname{Tr}(\mathbf{S}_i \mathbf{P}^{-1} \mathbf{S}_j \mathbf{P}^{-1}) \quad (4.9)$$

$$\|\mathbf{S}\|_{\mathbf{P}}^2 = \langle \mathbf{S}, \mathbf{S} \rangle_{\mathbf{P}} = \operatorname{Tr}(\mathbf{S} \mathbf{P}^{-1} \mathbf{S} \mathbf{P}^{-1}) \quad (4.10)$$

Note that for $\mathbf{P} = \mathbf{I}$, the identity matrix, this norm is equal to the squared vector norm or Frobenius norm:

$$\langle \mathbf{S}, \mathbf{S} \rangle_{\mathbf{I}} = \|\mathbf{S}\|_{\mathbf{F}}^2 = \sum_{i=1}^n \sum_{j=1}^n |s_{ij}|^2 = \operatorname{Tr}(\mathbf{S} \mathbf{S}^T), \quad (4.11)$$

where \mathbf{S} is a matrix of size $n \times n$, and s_{ij} elements of the \mathbf{S} matrix, with $1 \leq i \leq n$ and $1 \leq j \leq n$.

As illustrated in Figure 4.2, the geodesic is the shortest curve that connects two points on a manifold. Similarly, the Riemannian distance δ_R , between two SPD matrices P_i and P_j is given by the length of the geodesic, as shown in Figure 4.3. It can be computed as [5] [61] [63]:

$$\delta_R(P_i, P_j) = \|\log(P_i^{-1}P_j)\|_{\mathbf{F}} = \left[\sum_{i=1}^n \log^2 \lambda_i \right]^{1/2}, \quad (4.12)$$

where λ_i are the real eigenvalues of $P_i^{-1}P_j$.

4.3.1 Logarithmic and Exponential Map

Let $\gamma(t)$ be the geodesic between P and P_i . For each point $P_i \in \mathcal{P}(n)$ it is possible to identify a tangent vector $S_i \in \mathcal{S}(n)$. The logarithmic map, as it is showed in Figure 4.3, locally projects all the covariance matrices P_i into the tangent space:

$$S_i = \text{Log}_P(P_i) = P^{1/2} \log(P^{-1/2}P_i P^{-1/2}) P^{1/2} \quad (4.13)$$

while the exponential map allows to project an element from the tangent space back into the manifold:

$$P_i = \text{Exp}_P(S_i) = P^{1/2} \exp(P^{-1/2}S_i P^{-1/2}) P^{1/2}. \quad (4.14)$$

It turns out [5] that we can rewrite the Riemannian distance shown in Equation 4.12, by projecting the matrices P into the tangent space using the logarithmic map and applying the vectorization of a matrix shown in Equation 4.3 [5]:

$$\begin{aligned} \delta_R(P, P_i) &= \|\text{Log}_P(P_i)\| = \|S_i\|_P \\ &= \left\| \text{vect}(P^{-1/2} \text{Log}_P(P_i) P^{-1/2}) \right\|_2 \\ &\equiv \left\| \text{vect}(\tilde{S}_i) \right\|_2 \end{aligned} \quad (4.15)$$

If P_i is locally distributed into the manifold and P is the mean of all P_i , then it is possible to state that [5]:

$$\forall i, j \quad \delta_R(P_i, P_j) \approx \left\| \text{vect}(\tilde{S}_i) - \text{vect}(\tilde{S}_j) \right\|_2 \quad (4.16)$$

Hence, the projection into the tangent space allows us to approximation of the Riemannian distance using the Euclidean distance in tangent space. The following sections will discuss the advantage of projecting the SPD matrices into the tangent space to compute the geometric mean.

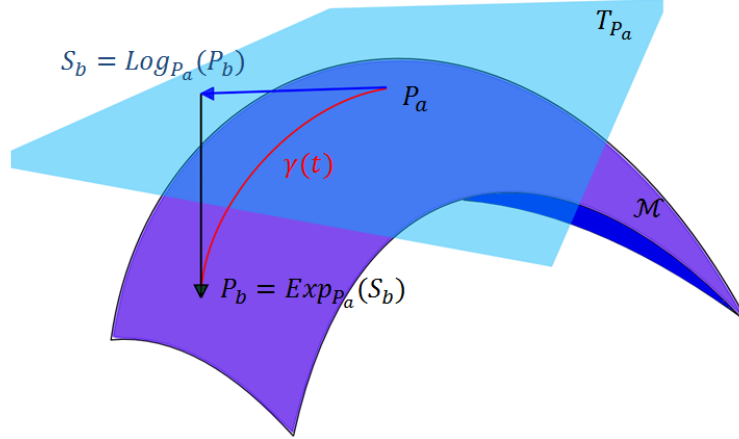


Figure 4.3: Manifold \mathcal{M} and the tangent space T_{P_a} at the point P_a . The red curve $\gamma(t)$ is the geodesic between P_a and P_b . The logarithmic map projects the point P_b into T_{P_a} with the relation: $S_b = \text{Log}_{P_a}(P_b)$. The exponential map projects the tangent vector S_b back in the manifold \mathcal{M} with the relation: $P_b = \text{Exp}_{P_a}(S_b)$

4.4 Means of SPD Matrices

Analogous to Euclidean space, the geometric mean of a set of points on the Riemannian manifold is the point on \mathcal{M} that minimizes the sum of the squared Riemannian distances δ_R [64]. Thus, the tangent space T related to the geometric mean gives the best local approximation for the projected points P from the manifold [59].

The following sections will explain how to compute the Arithmetic Mean and the Geometric Mean on a manifold.

4.4.1 Arithmetic Mean

For a set of I positive numbers and a vector x_i where $1 \leq i \leq I$. The arithmetic mean of x is defined as [63]:

$$\bar{x} = \frac{1}{I} \sum_{i=1}^I x_i \quad (4.17)$$

The concept of arithmetic mean can be extended to matrices. For a set of I matrices, $P_i \in \mathcal{P}(n)$, where $1 \leq i \leq I$, the arithmetic mean \mathfrak{A} , using Euclidean distance on the manifold \mathcal{M} is defined as [63] [5]:

$$\mathfrak{A}(P_1, \dots, P_I) = \arg \min_{P \in \mathcal{P}(n)} \sum_{i=1}^I \delta_E^2(P, P_i) = \frac{1}{I} \sum_{i=1}^I P_i, \quad (4.18)$$

where $\delta_E = \|P_1 - P_2\|_{\mathbf{F}}$ is the distance induced by the Frobenius norm.

4.4.2 Geometric Mean

Using Riemannian geodesic distance (Equation 4.12, Section 4.3) instead of Euclidian distance, the same concept can be applied to find the geometric mean. Given a set of I matrices $P_i \in \mathcal{P}(n)$ with $1 \leq i \leq I$ the geometric mean \mathfrak{G} can be defined as [5]:

$$\mathfrak{G}(P_1, \dots, P_I) = \arg \min_{P \in \mathcal{P}(n)} \sum_{i=1}^I \delta_R^2(P, P_i), \quad (4.19)$$

where δ_R uses the geodesic distance instead of Euclidean.

4.5 Computing the Mean on the Tangent Space

As shown in Section 4.3.1, the logarithmic map can be used to project the points from the Riemannian manifold into the tangent space. Since the tangent space is an Euclidean space, it is possible to compute the arithmetic mean \mathfrak{A} . Moreover, if the locality condition is respected, the relation in Equation 4.16 is valid allowing the computation of the geometric mean \mathfrak{G} by using the properties of the arithmetic mean \mathfrak{A} .

The iterative process to compute the geometric mean consists of the following steps [59].

1. Project the covariance matrices $P_i \in \mathcal{P}(n)$ into the tangent space $T_{\mathbf{P}}$ using the logarithmic map in Equation 4.13
2. Estimate the arithmetic mean \mathfrak{A} in the tangent space
3. Project \mathfrak{A} back into the manifold using the exponential map in Equation 4.14

These steps are repeated until convergence, as shown in Algorithm 1.

Algorithm 1: Geometric Mean of I SPD matrices

Data: $\epsilon > 0$ and a set of I SPD matrices $P_i \in \mathcal{P}(n)$ where $1 \leq i \leq I$

Result: \mathfrak{G} the estimated geometric mean in $\mathcal{P}(n)$

Initialization, $\mathfrak{G} = \frac{1}{I} \sum_{i=1}^I P_i$;

repeat

$\mathfrak{A}(P_1, \dots, P_I) = \frac{1}{I} \sum_{i=1}^I \text{Log}_{\mathfrak{G}}(P_i)$;

$\mathfrak{G} = \text{Exp}_{\mathfrak{G}}(\mathfrak{A})$;

until $\|\mathfrak{A}\|_F < \epsilon$;

return \mathfrak{G}

The geometric mean \mathfrak{G} , computed by Algorithm 1, is used as reference point, noted as P_{ref} . Since the Riemannian distance between two points is approximable to the Euclidean distance only locally, as stated in Equation 4.16, the geometric mean represents an appropriate reference point, given that on average it is at the smallest distance from all other points.

The tangent space is computed at P_{ref} in the manifold \mathcal{M} and all the P_i matrices are projected into this tangent space using the logarithmic map, shown in Equation 4.13 with P_{ref} as a base (see Section 5.1 of Chapter 5). Finally, the symmetric matrices S_i are vectorized, in order to be used as features for the regression models [59].

Thus, the projection of the covariance matrices into the tangent space enables the use of both traditional regression methods based on Euclidean distance and the Dual Representation of the regression based on the kernel method.

Chapter 5

Regression

Regression is a method used to predict the value of continuous variables y given the value of input vector variables x [65]. This chapter will describe the main theory of regression and in particular two regression learning methods, *Least Square Regression with Ridge Regularization* and *Dual Representation with Riemannian-based kernel*.

The regression process has two main steps, training and runtime. During the training step, the regression function, which predicts the known output y , is trained based on a dataset containing labeled data. Afterwards the trained model is used to predict the unknown target value y given an unlabeled data vector \mathbf{x} .

Given the training data $\{(x_1, y_1), \dots, (x_i, y_i), \dots, (x_I, y_I)\}$ with input data $x_i \in \mathbb{R}^n$, $1 \leq i \leq I$, and target values $y_i \in \mathbb{R}$. The regression method aims to find a function $f(x)$, $f : \mathbb{R}^n \rightarrow \mathbb{R}$, that predicts the target y_i , minimizing the error ε , as shown in Figure 5.1.

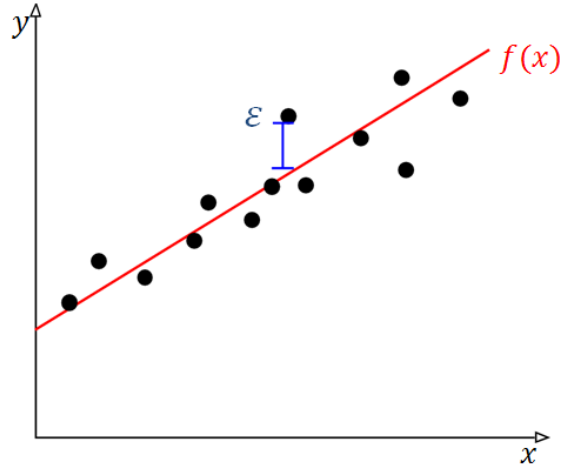


Figure 5.1: Example of linear regression. The points represent the input data. The red line represents the regression function $f(x)$ obtained by the regression analysis. The blue line represents the error ε , which is the distance between the predicted and actual value.

The function $f(\mathbf{x}, \mathbf{w})$ approximates the target variable y with an additive Gaussian noise ε that has a mean of zero and a variance equal to σ^2 , $p(\varepsilon) = \mathcal{N}(0, \sigma^2)$:

$$y = f(\mathbf{x}, \mathbf{w}) + \varepsilon \quad (5.1)$$

For a linear regression, the function f of Equation 5.1 can be defined as:

$$f(\mathbf{x}, \mathbf{w}) = w_0 + w_1x_1 + \dots + w_Ix_I, \quad (5.2)$$

where $\mathbf{w} = \{w_1, \dots, w_I\} \in \mathbb{R}^n$ is the weight vector, $w_0 \in \mathbb{R}$ is the bias and \mathbf{x} represents the input vector $\mathbf{x} = \{x_1, \dots, x_I\}^T$. The model written in Equation 5.2 is a linear function of weights and input variables.

In many cases there is not a linear relationship between the input data and the target data, which leads poor estimation of the targets. A possible solution is to transform the input data through a non-linear feature transformation, while f remains linear in \mathbf{w} [65].

This feature transformation is a map $\phi : \mathcal{X} \rightarrow \mathcal{F}$ where \mathcal{X} is the input space and \mathcal{F} is the feature space, as shown in Figure 5.2.

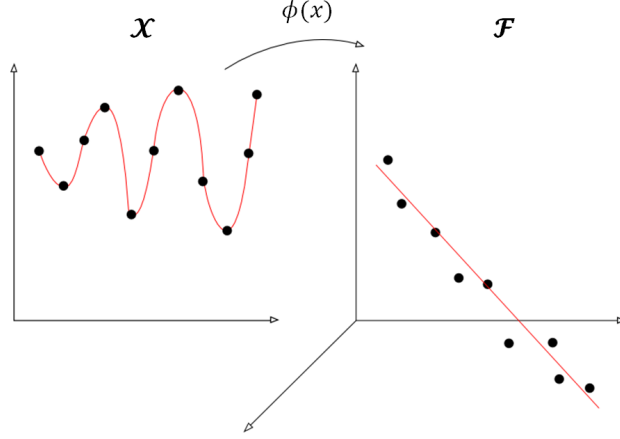


Figure 5.2: The left graph represents the input data in the input space \mathcal{X} . The right graph represents the input transformed into the feature space \mathcal{F} through the function ϕ . The nonlinear input data is linearized in the feature space.

Using a feature transformation it may be possible to apply linear regression models with success on input that is nonlinear in the original input space.

5.1 Our Feature Function

For this project the output or target variable y is given by the loudness or frequency levels defined in the experiments, while the input data is the symmetric positive definite (SPD) matrices P_i on the Riemannian manifold \mathcal{M} obtained from the brain's signals. First, the P_i matrices are mapped through the function [59]:

$$\phi(P_i) = \text{Log}_{P_{ref}}(P_i) = \bar{S}_i \quad (5.3)$$

The feature transformation in Equation 5.3, projects the P_i matrices into the tangent space at the reference point P_{ref} . Successively, the feature $\phi(P_i)$ is transformed to the normalized tangent space at P_{ref} , creating the symmetric matrix \tilde{S}_i :

$$\begin{aligned} \tilde{S}_i &= P_{ref}^{-1/2} \bar{S}_i P_{ref}^{-1/2} \\ &= P_{ref}^{-1/2} \text{Log}_{P_{ref}}(P_i) P_{ref}^{-1/2} \\ &= P_{ref}^{-1/2} P_{ref}^{1/2} \log(P_{ref}^{-1/2} P_i P_{ref}^{-1/2}) P_{ref}^{1/2} P_{ref}^{-1/2} \\ &= \log(P_{ref}^{-1/2} P_i P_{ref}^{-1/2}) \end{aligned} \quad (5.4)$$

Note that matrix \tilde{S}_i is used in Equations 4.15 and 4.16 for the distance computations between SPD matrices.

Then, to be able to apply the regression method, matrix \tilde{S}_i is vectorized (Equation 4.3). Thus, the regression method is applied on $\text{vect}(\tilde{S}_i)$. To simplify the notation in this chapter, we use $\phi(\mathbf{x}_i)$ to denote the vectorized feature vector.

In the model proposed in Equation 5.2, there is a linear relationship between every observed target y_i , $0 < i < I$ and $\phi_j(\mathbf{x}_i)$, $0 < j < P$ predictors, where P depends on the dimension of $\text{vect}(\tilde{S}_i)$.

$$y_i = w_0 + w_1\phi_1(\mathbf{x}_i) + \cdots + w_j\phi_j(\mathbf{x}_i) + \cdots + w_P\phi_P(\mathbf{x}_i) + \varepsilon \quad (5.5)$$

where $\phi_0 \equiv 1$.

Since this relationship is valid for every observation, Equation 5.5 can be rewritten in matrix form $\mathbf{y} = \mathbf{\Phi}\mathbf{w} + \mathbf{e}$, or equivalently [66]:

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} 1 & \phi_1(\mathbf{x}_1) & \phi_2(\mathbf{x}_1) & \cdots & \phi_P(\mathbf{x}_1) \\ 1 & \phi_1(\mathbf{x}_2) & \phi_2(\mathbf{x}_2) & & \vdots \\ \vdots & & & \ddots & \vdots \\ 1 & \phi_1(\mathbf{x}_I) & \cdots & \cdots & \phi_P(\mathbf{x}_I) \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_P \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_I \end{bmatrix} \quad (5.6)$$

where $\mathbf{y} = [y_1, \dots, y_I]^T$, $\mathbf{w} = [w_0, \dots, w_P]^T$, $\mathbf{e} = [\varepsilon_1, \dots, \varepsilon_I]^T$ and $\mathbf{\Phi}$ is a matrix of $I \times P$ dimension, whose elements are the features $\phi_j(\mathbf{x}_i)$, with $0 \leq j \leq P$ and $1 \leq i \leq I$.

5.2 Regression on the Feature Transformed Input

Using the feature function ϕ , the model in Equation 5.2, can be reformulated as:

$$f(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^P w_j \phi_j(\mathbf{x}) = \phi(\mathbf{x})\mathbf{w} \quad (5.7)$$

The ability of the model to predict the output depends on the dimension of ϕ and \mathbf{w} . High dimensionality creates a better fit for the training set but might cause errors in the runtime set. Figures 5.3, 5.4 and 5.5 show that increasing the number of parameters \mathbf{w} in the regression model creates a better fit for the training data.

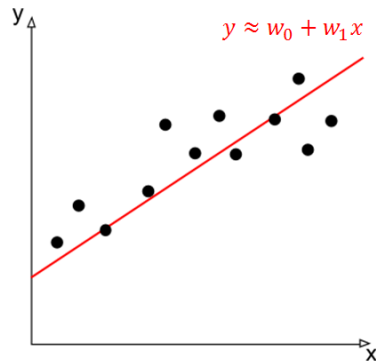


Figure 5.3: The function $f(x)$ illustrated by the red line approximates the output y of the training data given the input x and weights w .

In other words, a higher dimension of the weight variables w allows the regression function to be more representative of the training dataset.

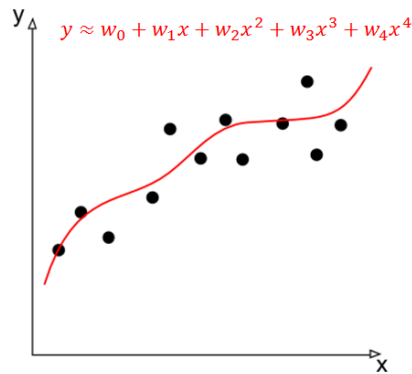


Figure 5.4: The function $f(x)$ approximates data better than the previous function, shown in Figure 5.3

Increasing the dimensionality of the predictors' reduces the error on training data. Nevertheless, it could cause large errors on the unlabeled data, shown in Figures 5.5 and 5.6. This phenomenon is known as overfitting and it is caused by the dependence of the function $f(x)$ on the structure of the training set, allowing large errors for the runtime set.

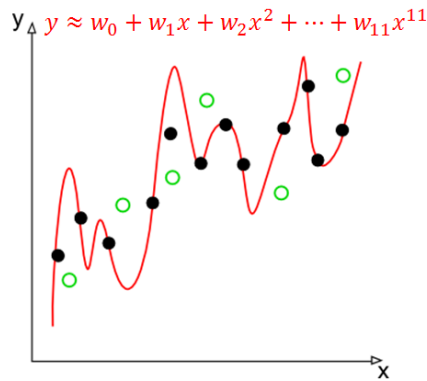


Figure 5.5: The function $f(x)$ affected by overfitting. The generated model is complex and dependent on the training set, which leads to large errors in the runtime set, represented by the green circles.

Thus, as showed in Figure 5.6, it is important to choose the parameters' dimension P appropriately in order to minimize the errors in both sets.

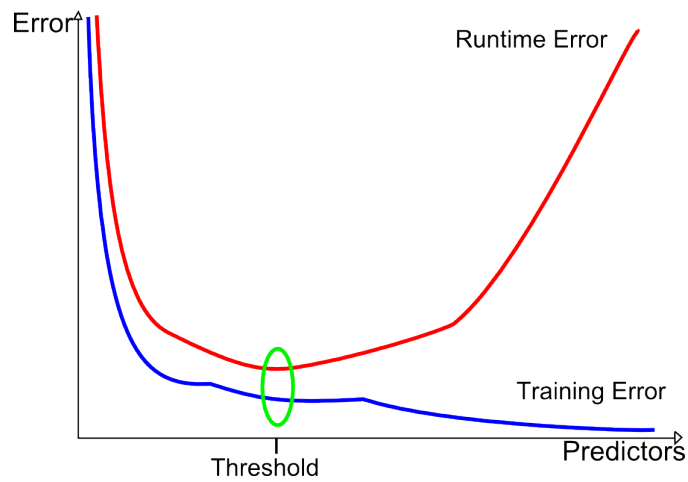


Figure 5.6: Variation of the prediction error in relation to the predictors' dimension. The blue line represents the Error on the training set, the red line the runtime error on a separate test set. The error of the training tends to decrease as we increase the predictors' dimension, while the error on the unlabeled runtime data tends to increase after a given threshold. It is important for the quality of the model to choose a number of predictors near to the threshold, that represent an appropriate model complexity.

5.3 Least Square Regression and Ridge Regularization

Given a set of input vectors $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_I\}$ and their corresponding targets $\mathbf{y} = \{y_1, \dots, y_I\}$ we can compute the weight vector \mathbf{w} of the linear regression function $f(\mathbf{x}, \mathbf{w})$ by minimizing the term $E_D(w)$, which is the squared sum of the difference between each target data and the estimation or output of the model [65]:

$$E_D(w) = \frac{1}{2} \sum_{i=1}^I (y_i - f(\mathbf{x}_i, \mathbf{w}))^2 \quad (5.8)$$

$E_D(w)$ is also denoted as the Least Squared Error. Thus Equation 5.8 will be minimized respect to \mathbf{w} :

$$\begin{aligned} \arg \min_{\mathbf{w}} E_D(w) &= \frac{1}{2} \sum_{i=1}^I (y_i - f(\mathbf{x}_i, \mathbf{w}))^2 \\ &= \frac{1}{2} \sum_{i=1}^I (y_i - \mathbf{w}^T \phi(\mathbf{x}_i))^2 \\ &= \frac{1}{2} ((\mathbf{y} - \Phi \mathbf{w})^T (\mathbf{y} - \Phi \mathbf{w})) \end{aligned} \quad (5.9)$$

The analytic solution to Equation 5.9 is found by setting the gradient with respect to \mathbf{w} to zero and solving the resulting equation:

$$\begin{aligned} \frac{\partial}{\partial \mathbf{w}} E_D(w) &= \frac{1}{2} \frac{\partial}{\partial \mathbf{w}} ((\mathbf{y} - \Phi \mathbf{w})^T (\mathbf{y} - \Phi \mathbf{w})) \\ &= \frac{1}{2} \frac{\partial}{\partial \mathbf{w}} ((\mathbf{y} - \Phi \mathbf{w})^T (\mathbf{y} - \Phi \mathbf{w})) \\ &= \frac{1}{2} ((\Phi \mathbf{w} - \mathbf{y})^T \Phi + \Phi^T (\Phi \mathbf{w} - \mathbf{y})) \\ &= \Phi^T \Phi \mathbf{w} - \Phi^T \mathbf{y} = 0 \end{aligned} \quad (5.10)$$

which implies that

$$\mathbf{w} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{y} \quad (5.11)$$

To control overfitting it is possible to regularize the $E_D(w)$ function in Equation 5.9 by adding a penalty term $E_w(w)$. The penalty term aims to reduce the feature dimension by pushing the weights to zero. Therefore the objective function becomes [65]:

$$\begin{aligned} J(\mathbf{w}) &= \arg \min_{\mathbf{w}} E_D(w) + \lambda E_w(w) \\ &= \frac{1}{2} ((\mathbf{y} - \Phi \mathbf{w})^T (\mathbf{y} - \Phi \mathbf{w})) + \lambda E_w(w) \end{aligned} \quad (5.12)$$

with $\lambda \geq 0$.

Two of the most used regularization terms are the Lasso (\mathcal{L}_1) regularization, where [67]:

$$E_w(w) = \sum_{i=0}^P |w_i| = |\mathbf{w}| \quad (5.13)$$

and the Ridge (\mathcal{L}_2) regularization [65], where:

$$E_w(w) = \frac{1}{2} \sum_{i=0}^P w_i^2 = \frac{1}{2} \mathbf{w}^T \mathbf{w} \quad (5.14)$$

\mathcal{L}_1 regularization is more suitable if there are many irrelevant features. In contrast, the Ridge regularization behaves better if most of the features are relevant. This project uses the \mathcal{L}_2 regularization, thus, the following theory uses this regularization term.

The objective function $J(\mathbf{w})$ for this case can be rewritten as shown in Equation 5.15, also known as Ridge Regression [68] [65]:

$$J(\mathbf{w}) = \arg \min_{\mathbf{w}} \frac{1}{2} ((\mathbf{y} - \Phi \mathbf{w})^T (\mathbf{y} - \Phi \mathbf{w})) + \frac{\lambda}{2} \mathbf{w}^T \mathbf{w} \quad (5.15)$$

As in the previous case, the solution of the Equation 5.15 is found by setting the gradient with respect to \mathbf{w} to zero.

$$\begin{aligned} \frac{\partial}{\partial \mathbf{w}} J(\mathbf{w}) &= \frac{\partial}{\partial \mathbf{w}} \left[\frac{1}{2} ((\mathbf{y} - \Phi \mathbf{w})^T (\mathbf{y} - \Phi \mathbf{w})) + \frac{\lambda}{2} \mathbf{w}^T \mathbf{w} \right] \\ &= \frac{1}{2} [\Phi^T (\Phi \mathbf{w} - \mathbf{y}) + (\Phi \mathbf{w} - \mathbf{y})^T \Phi] + \frac{\lambda}{2} \mathbf{w} \end{aligned} \quad (5.16)$$

Setting $\frac{\partial}{\partial \mathbf{w}} J(\mathbf{w}) = 0$ we obtain

$$\begin{aligned} \frac{1}{2} [\Phi^T (\Phi \mathbf{w} - \mathbf{y}) + (\Phi \mathbf{w} - \mathbf{y})^T \Phi] + \frac{\lambda}{2} \mathbf{w} &= 0 \\ \Phi^T \Phi \mathbf{w} - \Phi^T \mathbf{y} + \lambda \mathbf{w} &= 0 \\ \mathbf{w} (\Phi^T \Phi + \lambda I) &= \Phi^T \mathbf{y} \end{aligned} \quad (5.17)$$

which implies that

$$\mathbf{w} = (\Phi^T \Phi + \lambda I)^{-1} \Phi^T \mathbf{y} \quad (5.18)$$

Finally \mathbf{w} is substituted in the linear regression model defined in Equation 5.2 to predict the output given the input \mathbf{x} .

Thus, the training set is used to find the weight vector \mathbf{w} . Then the training set is discarded and the prediction on the new input \mathbf{x} is based only on \mathbf{w} [65].

There is another type of regression model that does not compute the weight vector \mathbf{w} for all the trials. This type of regression model is the Dual Representation of the regression function. This non-parametric technique uses the training set during the runtime step. The following section describes how to express the regression model in dual form, which allows the application of implicit feature transformations via the kernel trick.

5.4 Dual Representation with Riemannian-Based Kernel

A standard linear parametric model $f(\mathbf{x}, \mathbf{w})$ depends on the parameters \mathbf{w} . These weights are computed from the training dataset and are used to predict the targets in the runtime step. Thus the prediction of the target data in the runtime step depends only on \mathbf{w} .

Alternatively it is possible to reformulate the regression model in terms of a dual representation that allows to use the data directly. This is accomplished by using a kernel function given by the relation [65]:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j) \quad (5.19)$$

The kernel function used in this project is the Riemannian-based kernel. Considering the input vectors as points P_i in the manifold \mathcal{M} , the Riemannian-based kernel is defined by combining the Equations 5.3 and 4.9 to obtain [59]:

$$\begin{aligned} k_R(\text{vect}(P_i), \text{vect}(P_j); P_{ref}) &= \langle \phi(P_i), \phi(P_j) \rangle_{P_{ref}} \\ &= Tr \left[\text{Log}_{P_{ref}}(P_i) P_{ref}^{-1} \text{Log}_{P_{ref}}(P_j) P_{ref}^{-1} \right] \\ &= Tr \left[\log(P_{ref}^{-1/2} P_i P_{ref}^{-1/2}) \log(P_{ref}^{-1/2} P_j P_{ref}^{-1/2}) \right], \end{aligned} \quad (5.20)$$

Equation 5.20 can be rewritten as [59]:

$$\begin{aligned} k_R(\text{vect}(P_i), \text{vect}(P_j); P_{ref}) &= Tr \left[P_{ref}^{-1/2} \text{Log}_{P_{ref}}(P_i) P_{ref}^{-1/2} P_{ref}^{-1/2} \text{Log}_{P_{ref}}(P_j) P_{ref}^{-1/2} \right] \\ &= \left\langle \tilde{S}_i, \tilde{S}_j \right\rangle_{\mathbf{F}} \\ &= \text{vect}(\tilde{S}_i)^T \text{vect}(\tilde{S}_j) \end{aligned} \quad (5.21)$$

Now, the solution of the linear regression model in Equation 5.15 can be rewritten as:

$$\mathbf{w} = -\frac{1}{\lambda} \sum_{i=1}^I (\mathbf{w}^T \phi(\mathbf{x}_i) - y_i) \phi(\mathbf{x}_i) = \sum_{i=1}^I a_i \phi(\mathbf{x}_i), \quad (5.22)$$

where $a_i = \frac{1}{\lambda} \sum_{i=1}^I (\mathbf{w}^T \phi(\mathbf{x}_i) - y_i)$.

In matrix form, Equation 5.22 can be expressed as

$$\mathbf{w} = \Phi^T \mathbf{a}, \quad (5.23)$$

where the vector $\mathbf{a} = (a_1, \dots, a_I)$.

Substituting $\mathbf{w} = \Phi^T \mathbf{a}$ into Equation 5.12, we can now redefine the regression model in the Dual Representation, based on the vector \mathbf{a} [65]:

$$J(\mathbf{a}) = \frac{1}{2} \mathbf{a}^T \Phi \Phi^T \Phi \Phi^T \mathbf{a} - \mathbf{a}^T \Phi \Phi^T \mathbf{y} + \frac{1}{2} \mathbf{y}^T \mathbf{y} + \frac{\lambda}{2} \mathbf{a}^T \Phi \Phi^T \mathbf{a} \quad (5.24)$$

Let us define the matrix $\mathbf{K} = \Phi \Phi^T$. This matrix, called the Gram matrix, is a symmetric $I \times I$ matrix whose elements $K_{i,j}$ are given by the kernel function:

$$K_{i,j} = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j) = k(\mathbf{x}_i, \mathbf{x}_j) \quad (5.25)$$

Thus, the objective function can be written as:

$$J(\mathbf{a}) = \frac{1}{2} \mathbf{a}^T \mathbf{K} \mathbf{K} \mathbf{a} - \mathbf{a}^T \mathbf{K} \mathbf{y} + \frac{1}{2} \mathbf{y}^T \mathbf{y} + \frac{\lambda}{2} \mathbf{a}^T \mathbf{K} \mathbf{a} \quad (5.26)$$

It is possible to find the solution by evaluating the gradient of Equation 5.26 with respect to \mathbf{a} and setting it to zero, obtaining:

$$\mathbf{a} = (\mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{y} \quad (5.27)$$

where \mathbf{I} is the identity matrix of size $I \times I$.

Let us define the column vector $\mathbf{k}(\mathbf{x})$ with I elements $k_i(\mathbf{x}) = k(\mathbf{x}_i, \mathbf{x})$, $1 \geq i \leq I$. By substituting the vector parameter \mathbf{a} found in the linear regression model of Equation 5.2 we obtain the prediction for an input \mathbf{x} [65]:

$$f(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) = \mathbf{a}^T \Phi \phi(\mathbf{x}) = \mathbf{k}(\mathbf{x})^T (\mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{y} \quad (5.28)$$

The Dual Representation presented in this section produces the same results as the Ridge Regularization model, since an explicit feature transformation, $\phi(\mathbf{x}_i) = \text{Log}_{P_{ref}} P_i$, is used.

It is possible to construct any other type of kernel on top of the presented model, for example a Gaussian kernel [65]. Since this is a preliminary study, during the implementation of the project we chose not to apply any further transformation on top of the *Riemannian-based kernel*. Nevertheless, the following section presents how to apply a Gaussian kernel on top of the Riemannian-based kernel.

5.5 Constructing a Gaussian Kernel

The Dual Representation based on the kernel method, as seen in Section 5.4, is not computationally convenient with respect to the Least Squared Regression. The non-parametric regression method computes the regression function by inverting the Gram Matrix of dimension $I \times I$, while the computation of the weights' vector \mathbf{w} is based on the inversion of the $\Phi^T \Phi$ matrix of dimension $P \times P$. Because generally the number of trials $I \gg P$, the Dual Representation does not seem to be a convenient method [65].

Nevertheless, this computational inefficiency originates from the explicit data transformation $\phi(\mathbf{x})$, and expressing the Dual Representation only using the kernel function will avoid the explicit transformation. Thus, the kernel representation allows to implicitly use a feature space of high dimensionality or eventually applying a new kernel function on top of an existing one [65] [69]. This last property is particularly attracting since it allows to implicitly transform the data in a new feature space.

Let a valid kernel, as the Riemannian-based kernel, be given. The Riemannian-based kernel is a valid kernel since it respects the condition in Equation 5.19 or alternatively since the Gram Matrix, generated from the Riemannian-based kernel, is a positive semi-definite matrix.

It is possible to apply another valid kernel on top of the Riemannian one, for example, the Gaussian kernel defined as:

$$k_G(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right) \quad (5.29)$$

Using the following kernel properties [65]:

- Given a valid kernel $k(\mathbf{x}_i, \mathbf{x}_j)$ and any function h , $k_1 = h(\mathbf{x}_i)k(\mathbf{x}_i, \mathbf{x}_j)h(\mathbf{x}_j)$ is a valid kernel.
- Given a valid kernel $k(\mathbf{x}_i, \mathbf{x}_j)$, $k_1 = \exp(k(\mathbf{x}_i, \mathbf{x}_j))$ is a valid kernel.

and the expansion of the square $\|\mathbf{x}_i - \mathbf{x}_j\|^2 = \mathbf{x}_i^T \mathbf{x}_i - 2\mathbf{x}_i^T \mathbf{x}_j + \mathbf{x}_j^T \mathbf{x}_j$, it is possible to verify that the Gaussian kernel is a valid kernel:

$$k_G(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\mathbf{x}_i^T \mathbf{x}_i}{2\sigma^2}\right) \exp\left(\frac{\mathbf{x}_i^T \mathbf{x}_j}{\sigma^2}\right) \exp\left(-\frac{\mathbf{x}_j^T \mathbf{x}_j}{2\sigma^2}\right) \quad (5.30)$$

As it is possible to see from the relation in Equation 5.19, the elements $\mathbf{x}^T \mathbf{x}$ represent a valid kernel, thus the Gaussian kernel can be rewritten as:

$$k_G(\mathbf{x}_i, \mathbf{x}_j) = \exp\left\{-\frac{1}{2\sigma^2}(k(\mathbf{x}_i, \mathbf{x}_i) + k(\mathbf{x}_j, \mathbf{x}_j) - 2k(\mathbf{x}_i, \mathbf{x}_j))\right\} \quad (5.31)$$

In this way it is possible to apply a Gaussian kernel on top of a valid kernel k . The kernel k inside the exponential function in Equation 5.31 can be substituted with the Riemannian-based kernel $k_R(\text{vect}(\mathbf{P}_i), \text{vect}(\mathbf{P}_j); \mathbf{P}_{ref}) = \text{vect}(\tilde{\mathbf{S}}_i)^T \text{vect}(\tilde{\mathbf{S}}_j)$.

Chapter 6

Estimation Of Covariance Matrices

By applying a linear regression method, as shown in Figure 6.1, it is possible to estimate the loudness or frequency level for a given BCI covariance input.

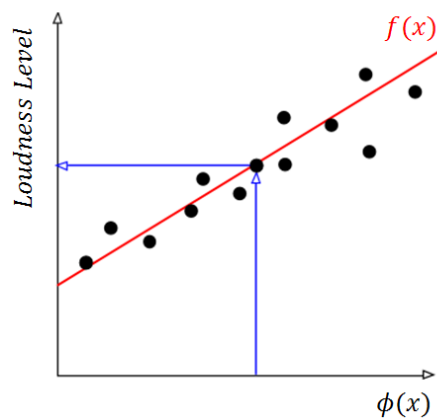


Figure 6.1: The function $f(x)$ estimates the loudness level of a given input $\phi(x)$.

Nevertheless, the purpose of this project includes also the inverse calculation, namely to be able to estimate the mental state of a user given a particular loudness or frequency level. The following sections discuss the interpolation method that enables the estimation of the covariance matrix P given a target point y .

6.1 Interpolation

The interpolation is a mathematical method that allows us to find new data points from a limited set of known data points. Therefore in this project, interpolation can be used to estimate the covariance matrix of any frequency or loudness level from a given set of covariance matrices. To contextualize the interpolation of covariance matrices, we first consider interpolation in Euclidean space and then progress into the interpolation in the appropriate Riemannian space.

6.1.1 Interpolation in the Euclidean Space

Let us define the subtraction between two points P_i and P_j in Euclidean space as $\overrightarrow{P_i P_j} = P_j - P_i$, shown in Figure 6.2

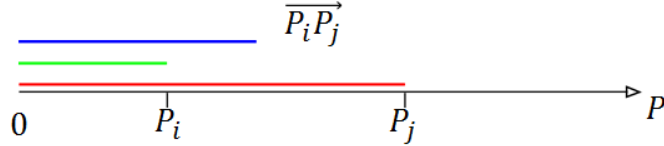


Figure 6.2: Two points in the Euclidean space. The green line denotes the position of P_i with respect to 0 and the red line the position of P_j . The blue line identifies the subtraction of $P_j - P_i$.

It is possible to estimate the position of a point P_r , between P_i and P_j , using linear interpolation [70].

$$\hat{P}_r = P_i + \gamma \overrightarrow{P_i P_j} \quad (6.1)$$

with $\gamma \in [0, 1]$. As shown in Figure 6.3.

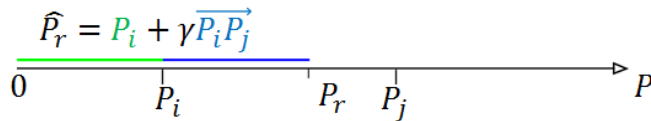


Figure 6.3: Estimation of the point P_r , between P_i and P_j , through the interpolation method. The green line identifies P_i while the blue line represent the subtraction between P_j and P_i conveniently shrunk by the γ parameter .

6.1.2 Interpolation in the Riemannian Manifold

Correspondingly, Riemannian Interpolation aims to estimate the value of a variable that falls in between two measurements in a Riemannian manifold.

As explained in Chapter 4, given two Covariance Matrices, P_i and P_j , laying in the Riemannian manifold, it is possible to compute the tangent vector S_j related to the geodesic between P_i and P_j by applying [70]:

$$S_j = \text{Log}_{P_i}(P_j) \quad (6.2)$$

Compared to Euclidean space, S_j represents the subtraction between P_j and P_i : $\overrightarrow{P_i P_j} = P_j - P_i$. For notational purposes we use $\overrightarrow{P_i P_j}$ to denote S_j .

It is possible to estimate the position of a Sample Covariance Matrix \hat{P}_r based on the Riemannian Linear Interpolation between the two covariance matrices P_i and P_j [70]. This method applies the interpolation in the tangent space and then projects it back into the Riemannian manifold:

$$\hat{P}_r = \text{Exp}_{P_i}(\gamma \overrightarrow{P_i P_j}), \quad (6.3)$$

with $\gamma \in [0, 1]$ and Exp_{P_i} being the exponential map.

6.1.3 Interpolation In Our Project

In this project the Riemannian interpolation method is used to estimate the Covariance Matrices at any possible frequency or loudness value. Let the input set $P(n) = \{P_1, \dots, P_I\}$ of Covariance Matrices be defined for the observation set $\mathbf{y} = \{y_1, \dots, y_I\}$.

Given an observation y_i , it is possible to find the related inputs P_i . There could be multiple P_i related to the observation y_i as shown in Figure 6.4. For example in the frequency experiment there are 80 matrices P related to each target frequency y .

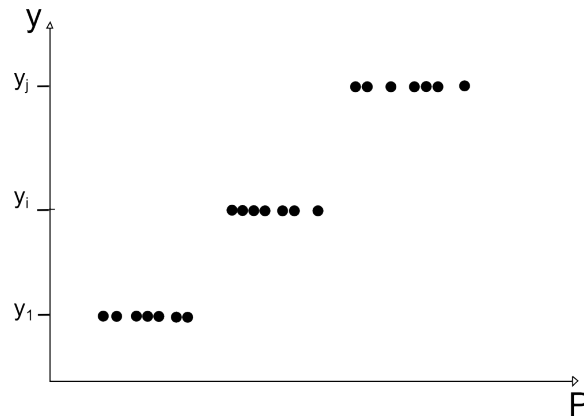


Figure 6.4: The black dots represent the covariance matrices P related to the same target y .

Therefore, for each y_i the geometric mean P_{mean_i} of the P matrices related to that target value has been computed using the Algorithm 1 in Chapter 4, as shown in Figure 6.5.

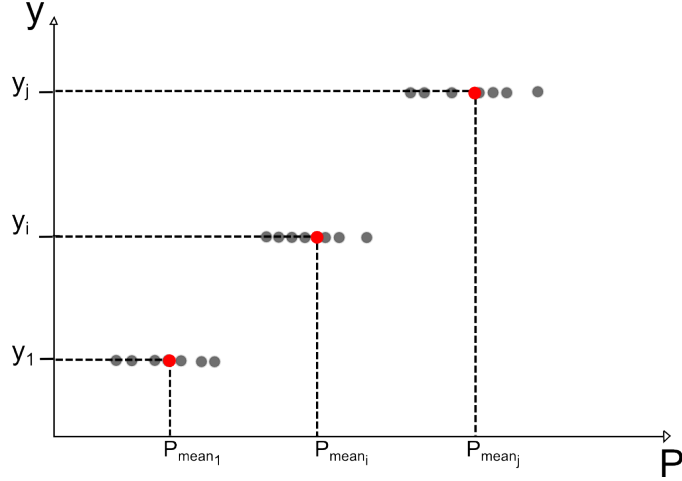


Figure 6.5: The red dots represent the geometric mean between the P matrices of three target values y_1 , y_i and y_j

It is possible to estimate the P_r covariance matrix of a new unobserved point y_r that falls between two observations y_i and y_j . Using the geometric mean, as seen previously, each of these two observations y_i and y_j has a related geometric mean, P_{mean_i} and P_{mean_j} , respectively.

It is possible to apply the Logarithmic map between P_{mean_i} and P_{mean_j} , as seen in Equation 6.2:

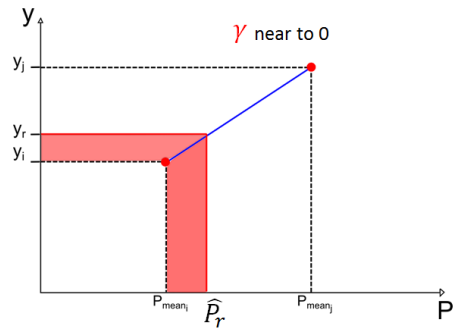
$$S_j = \text{Log}_{P_{\text{mean}_i}}(P_{\text{mean}_j}) \quad (6.4)$$

Finally, we can estimate the covariance matrix P_r related to y_r by applying the Riemannian Interpolation as seen in Equation 6.3:

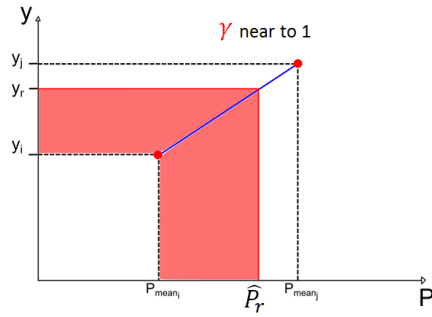
$$\hat{P}_r = \text{Exp}_{P_{\text{mean}_i}}(\overrightarrow{\gamma P_{\text{mean}_i} P_{\text{mean}_j}}) \quad (6.5)$$

Note that it is necessary to chose the γ parameter carefully. If the locality condition is respected and the relation in Equation 4.16 is valid, γ depends from the position of y_r respects to y_i and y_j , as shown in Figure 6.6:

$$\gamma = \frac{y_r - y_i}{y_j - y_i} \quad (6.6)$$



(a) γ computed from Equation 6.6 is small.



(b) γ computed from Equation 6.6 is large

Figure 6.6: The value of γ is related to the position of y_r .

Note that it is possible to apply the interpolation method only after that a valid linear regression is found. If there is not a linear regression function, the interpolation method estimates the wrong covariance matrices, as shown in Figure 6.7.

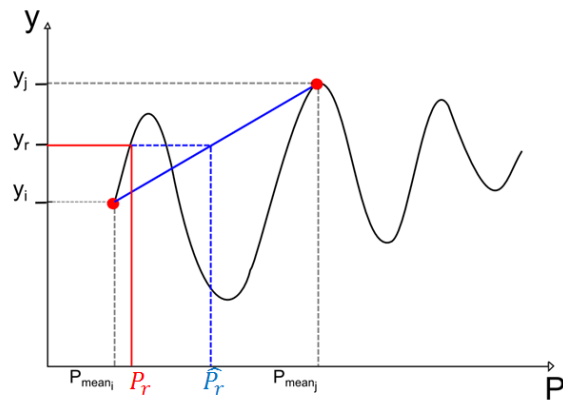


Figure 6.7: An interpolation method applied on covariance matrices without a linear relationship. The estimated point \hat{P}_r , represented in blue, is far from the actual value of P_r , in red.

Part III

Sound Parameter Variation Analysis

Overview of the Analysis

This chapter will give a description of the steps developed to analyze the information obtained through the *loudness* and *frequency* experiments. Figure 6.8 shows the procedure divided in four main steps.

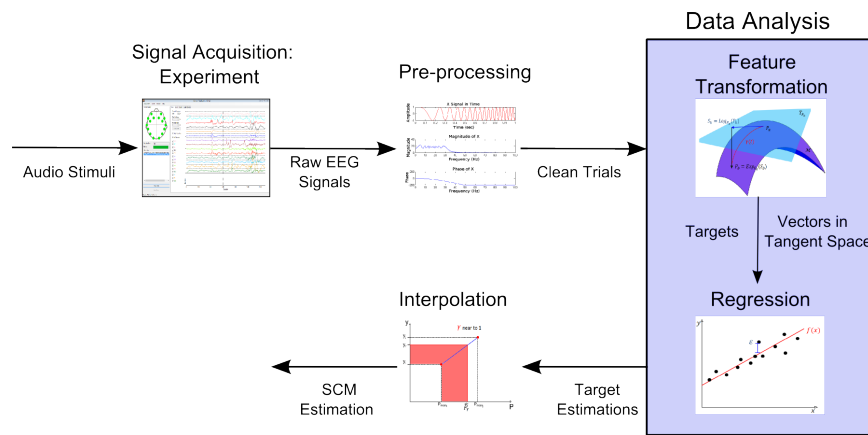


Figure 6.8: The 4 main steps that compose the analysis of the data.

1. **Signal Acquisition:** During the experiments, each subject is exposed to audio stimuli while the Emotiv EPOC records his/her EEG signals. The raw signals are then saved to an external file using the TestBench software. Chapter 7 gives a detailed description of the experiments.
2. **Pre-Processing:** The raw EEG signals from each subject are cut into trials. Only trials that contain no head movement, blinks, connection problems or low contact quality are selected. The trials are then filtered and specific brain rhythms are chosen. Chapter 8 describes the implementation for this step.
3. **Data Analysis**
 - **Feature Transformation:** Each trial is used to compute the Sample Covariance Matrices. The geometric mean is then calculated using the theory explained in Chapter 4. The features are

obtained from the vectorization of the symmetric matrices \tilde{S}_i in the tangent space. The implementation is described in Chapter 9.

- **Regression:** The transformed features are used as predictors for the Least Square Regression with Ridge Regularization and Dual Representation with Riemannian Based-kernel methods, as explained in Chapter 5. A description of both regressions' implementation can be found in Chapter 9.
4. **Interpolation:** Once a linear regression has been found it is possible to apply the Interpolation method to estimate the covariance matrices of any given *frequency* or *loudness* level between the minimum and maximum observed values, as shown in Chapter 6. The implementation for the interpolation method is shown in Chapter 10.

Chapter 7

Signal Acquisition: Experiments

This chapter presents the implementation of the *Signal Acquisition* phase of the project, namely the experiments. Figure 6.8 shows an overview of the experiment's development.

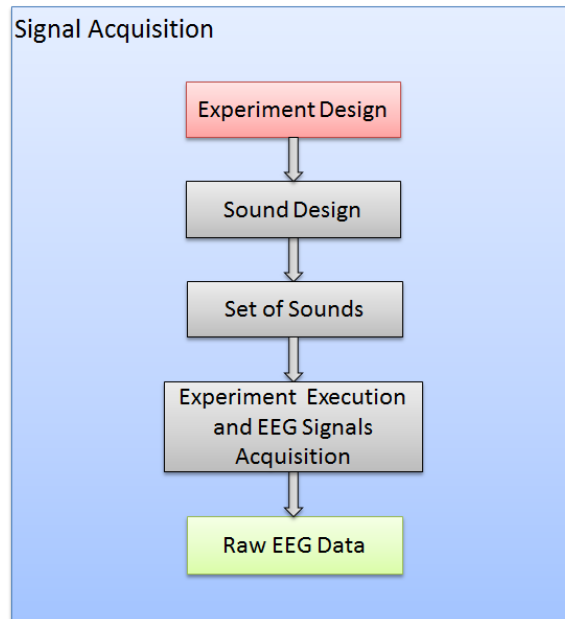


Figure 7.1: Overview of the Signal Acquisition phase. Both Loudness Experiment and Frequency Experiment follow this pipeline. The raw EEG data obtained from the experiments is used in the next phase of the analysis, Pre-processing.

The experiments created for this project were constructed in order to find a statistical linear regression, on Riemannian space, from the state of the listener's brain to specific sound parameter variations. They were designed to expose the listener to different sounds, measuring the brain's reaction through an EEG device and recording all the information.

7.1 Experiment Design

As explained in Chapter 2, a sound is comprised of several parameters that change the construction of the sound and the way we hear it. All sounds have a *fundamental frequency* and it usually is the predominant, most discernible sound in the combination of partials. Nevertheless, our perception of the sound is strongly influenced by the rest of the partials, every combination of harmonics and overtones create a particular sound. Human perception of *fundamental frequency* is also strongly influenced by the ADSR envelope. For example, a sound with very fast attack and short sustain is not perceived to have a clear pitch, like drums or non-tonal percussions. All sounds also have a volume or *loudness*, and this parameter can vary from imperceptible to painful, causing the perception of the *fundamental frequency* and timbre

to change.

The relationship between the sound parameters is nonorthogonal. Therefore, in order to understand how sound affects the human brain it is necessary to simplify and observe the main effect of a single parameter. If single parameter variation is understood for all sound parameters then the effect of combinations of parameters can be inferred and studied with a better understanding. In order to obtain information concerning a single parameter, all other parameters need to be static.

For this project two parameters were chosen, *loudness* and *frequency*. This selection was based on choosing simple parameters that have clear, perceptible distinction when the level is changed. Also, changes in *loudness* or *frequency* can be perceived without affecting drastically the perception of the other parameters. Another reason for selecting these two parameters is that there is research showing an evident reaction to both *loudness* and *frequency* changes in music, games and audio. Therefore we could expect the EEG signals to show a response to these parameters.

The following section elaborates on the contents and procedures followed for the two experiments.

7.1.1 Loudness Experiment

This experiment was designed to measure the brain's reaction to changes in the volume of a sound. Therefore, *loudness* is the variable parameter, while frequency, timbre, and ADSR are fixed.

Frequency is limited to three values, each from a different range: low, middle and high. The selection of the frequencies was based on a musical chromatic scale, the notes are separated by 33 semitones from each other. The three notes, D2, B4 and G#7 correspond to the following frequencies: 73.416Hz, 493.883Hz and 3322.44Hz respectively.

Timbre is fixed to four values, contrastingly different from each other. The four selected timbres are: scream, choir, wind instrument, and pure sinusoidal wave. The selection was based on covering both comfortable and uncomfortable, natural, and artificial sounds. We define a natural sound as a sound that can be produced by non-digital sources. Table 7.1 shows the classification of the selected timbres.

	Comfortable	Uncomfortable
Natural	B4-Choir	B4-Scream
	D2-Wind Instrument	G#7-Choir
	B4-Wind Instrument	G#7-Wind Instrument
Artificial	D2-Choir	D2-Sine Wave
		B4-Sine Wave
		G#7-Sine Wave
		D2-Scream
		G#7-Scream

Table 7.1

The ASDR envelope of each sound is the original natural envelope of their corresponding timbre.

Loudness is set to 9 different values and then applied to each of the twelve sounds (4 timbres \times 3 frequencies). The selected loudness levels range between 0dBFS and -48dBFS with a step of -6 dBFS. Note that dB Full Scale is used instead of dB for the digitally generated sounds.

Every sound lasts five seconds followed by three seconds of silence to separate each sound from each other clearly and let the subject relax.

Table 7.2, synthesizes the experiment's setup.

Frequency	Timbre	Loudness
36.7081Hz (D2)	Scream	0dBFS
246.942Hz (B4)	Choir	-6dBFS
1661.22Hz (G#7)	Sine Wave	-12dBFS
	Wind Instrument	-18dBFS
		-24dBFS
		-30dBFS
		-36dBFS
		-42dBFS
		-48dBFS

Table 7.2

7.1.2 Frequency Experiment

This experiment was designed to measure the brain’s reaction to changes in the *frequency* of a sound. Therefore the only variable sound is *frequency* while loudness, timbre and ADSR are fixed. Note that sounds can vary from a simple sine wave to complex sounds with several partials. In this experiment we understand *frequency* as the *fundamental frequency* of the sound, all the selected sounds have a *fundamental frequency* that is clear and distinguishable from all the partials.

The *frequency* variation selected for the experiment is based on the musical chromatic scale. Three tones were selected per octave, every tone separated from each other by four semitones. The range of selected frequencies is $\{C1, E1, G\#1, C2, \dots, E7, G\#7, C8\}$, their frequency in Hz is shown in Table 7.3.

Timbre is fixed to four values, all with an ADSR envelope that allows the production of a clear, sustained *frequency*. The four selected timbres are: piano, harp and two different virtual sounds 303FM8 and Embracer. The ADSR envelope of each sound is the original natural envelope of their corresponding timbre. Loudness was set to 0dBFS. Thus, the experiment has been conducted for each of these timbres.

Every sound lasts five seconds followed by three seconds of silence to separate each sound from each other clearly and let the subject relax.

A synthesis of the experiment’s setup is shown in Table 7.3.

Frequency	Timbre	Loudness
32.7032Hz (C1) 415.305Hz (G#4)	Piano	0dBFS
41.2034Hz (E1) 523.251Hz (C5)	Harp	
51.9131Hz (G#1) 659.255Hz (E5)	303FM8	
65.4064Hz (C2) 830.609Hz (G#5)	Embracer	
82.4069Hz (E2) 1046.50Hz (C6)		
103.826Hz (G#2) 1318.51Hz (E6)		
130.813Hz (C3) 1661.22Hz (G#6)		
164.814Hz (E3) 2093.00Hz (C7)		
207.652Hz (G#3) 2637.02Hz (E7)		
261.626Hz (C4) 3322.44Hz (G#7)		
329.628Hz (E4) 4186.01Hz (C8)		

Table 7.3

All the sounds, except for the scream, were generated using virtual instruments (VST). The scream file was obtained online, and is royalty free. All audio files were exported in WAV format with a sampling frequency of 44100Hz and bit-rate of 2822kbps.

7.2 Experiment Implementation

A program was designed to play the sounds in random order. The program controls the synchronization with the TestBench for recording and saving of all EEG readings from the Emotiv EPOC headset.

Afterwards, another program uses the CSV from the TestBench to cut the data into trials and sort them by subject. Then it eliminates the trials that have low contact quality, sudden head movements, blinks or connection problems. Finally, it saves the matrix containing the trials and the vector with the *frequency* or *loudness* target values into files ready to load in MATLAB.

7.3 Experiment Execution

The experiments were executed in a small sound-isolated room of approximately 25 m^2 . Each subject was placed in an arm chair, facing two Bowers&Wilkins DM7 MKII speakers [71]. The speakers and chair were placed as vertices of an equilateral triangle, each side with a length of 115cm. A

total of 20 subjects volunteered for the experiments. The age of the subjects ranges between 12 and 55 years old, with an average age of 30 years. All subjects are healthy and do not have any hearing disabilities.

The experiments were conducted over two days, the hardware setup and environment were maintained as stable as possible. The loudness of the both experiments was measured using a dB meter placed at ear height in the position of the subject. For the *frequency* experiment all the sounds' loudness levels fall between 75dB and 86dB with no drastic loudness changes between sounds. For the *loudness* experiment the measurement of each sound is shown Table 7.4.

Sound	Loudness	Sound	Loudness
D2-WindInstrument (-0dBFS)	102.3dB	D2-Choir (-0dBFS)	96.9dB
D2-WindInstrument (-6dBFS)	97.0dB	D2-Choir (-6dBFS)	90.2dB
D2-WindInstrument (-12dBFS)	90.7dB	D2-Choir (-12dBFS)	83.3dB
D2-WindInstrument (-18dBFS)	84.5dB	D2-Choir (-18dBFS)	79.1dB
D2-WindInstrument (-24dBFS)	79.7dB	D2-Choir (-24dBFS)	73.0dB
D2-WindInstrument (-30dBFS)	73.7dB	D2-Choir (-30dBFS)	67.3dB
D2-WindInstrument (-36dBFS)	67.7dB	D2-Choir (-36dBFS)	62.8dB
D2-WindInstrument (-42dBFS)	61.8dB	D2-Choir (-42dBFS)	55.6dB
D2-WindInstrument (-48dBFS)	56.0dB	D2-Choir (-48dBFS)	53.6dB
B4-WindInstrument (-0dBFS)	104.2dB	B4-Choir (-0dBFS)	105.6dB
B4-WindInstrument (-6dBFS)	98.3dB	B4-Choir (-6dBFS)	100.2dB
B4-WindInstrument (-12dBFS)	92.6dB	B4-Choir (-12dBFS)	93.6dB
B4-WindInstrument (-18dBFS)	86.2dB	B4-Choir (-18dBFS)	86.3dB
B4-WindInstrument (-24dBFS)	81.2dB	B4-Choir (-24dBFS)	83.8dB
B4-WindInstrument (-30dBFS)	75.4dB	B4-Choir (-30dBFS)	77.9dB
B4-WindInstrument (-36dBFS)	69.2dB	B4-Choir (-36dBFS)	71.8dB
B4-WindInstrument (-42dBFS)	63.2dB	B4-Choir (-42dBFS)	65.4dB
B4-WindInstrument (-48dBFS)	57.4dB	B4-Choir (-48dBFS)	59.7dB
G#7-WindInstrument (-0dBFS)	97.4dB	G#7-Choir (-0dBFS)	101.0dB
G#7-WindInstrument (-6dBFS)	91.6dB	G#7-Choir (-6dBFS)	95.5dB
G#7-WindInstrument (-12dBFS)	84.4dB	G#7-Choir (-12dBFS)	88.7dB
G#7-WindInstrument (-18dBFS)	81.3dB	G#7-Choir (-18dBFS)	82.4dB
G#7-WindInstrument (-24dBFS)	78.0dB	G#7-Choir (-24dBFS)	80.6dB
G#7-WindInstrument (-30dBFS)	75.1dB	G#7-Choir (-30dBFS)	73.2dB
G#7-WindInstrument (-36dBFS)	64.7dB	G#7-Choir (-36dBFS)	66.9dB
G#7-WindInstrument (-42dBFS)	60.0dB	G#7-Choir (-42dBFS)	64.0dB
G#7-WindInstrument (-48dBFS)	54.7dB	G#7-Choir (-48dBFS)	53.7dB

Sound	Loudness	Sound	Loudness
D2-Scream (-0dBFS)	109.0dB	D2-SineWave (-0dBFS)	96.1dB
D2-Scream (-6dBFS)	102.9dB	D2-SineWave (-6dBFS)	90.0dB
D2-Scream (-12dBFS)	96.3dB	D2-SineWave (-12dBFS)	81.1dB
D2-Scream (-18dBFS)	89.3dB	D2-SineWave (-18dBFS)	75.5dB
D2-Scream (-24dBFS)	82.2dB	D2-SineWave (-24dBFS)	69.3dB
D2-Scream (-30dBFS)	79.6dB	D2-SineWave (-30dBFS)	62.5dB
D2-Scream (-36dBFS)	73.5dB	D2-SineWave (-36dBFS)	56.1dB
D2-Scream (-42dBFS)	67.4dB	D2-SineWave (-42dBFS)	52.4dB
D2-Scream (-48dBFS)	61.0dB	D2-SineWave (-48dBFS)	48.6dB
B4-Scream (-0dBFS)	109.6dB	B4-SineWave (-0dBFS)	109.4dB
B4-Scream (-6dBFS)	104.9dB	B4-SineWave (-6dBFS)	103.5dB
B4-Scream (-12dBFS)	98.9dB	B4-SineWave (-12dBFS)	97.6dB
B4-Scream (-18dBFS)	92.3dB	B4-SineWave (-18dBFS)	91.6dB
B4-Scream (-24dBFS)	85.7dB	B4-SineWave (-24dBFS)	85.5dB
B4-Scream (-30dBFS)	81.7dB	B4-SineWave (-30dBFS)	82.5dB
B4-Scream (-36dBFS)	75.8dB	B4-SineWave (-36dBFS)	77.1dB
B4-Scream (-42dBFS)	69.8dB	B4-SineWave (-42dBFS)	68.0dB
B4-Scream (-48dBFS)	63.4dB	B4-SineWave (-48dBFS)	62.2dB
G#7-Scream (-0dBFS)	105.2dB	G#7-SineWave (-0dBFS)	110.5dB
G#7-Scream (-6dBFS)	99.3dB	G#7-SineWave (-6dBFS)	104.3dB
G#7-Scream (-12dBFS)	92.6dB	G#7-SineWave (-12dBFS)	98.9dB
G#7-Scream (-18dBFS)	86.0dB	G#7-SineWave (-18dBFS)	92.3dB
G#7-Scream (-24dBFS)	79.8dB	G#7-SineWave (-24dBFS)	85.6dB
G#7-Scream (-30dBFS)	75.8dB	G#7-SineWave (-30dBFS)	79.6dB
G#7-Scream (-36dBFS)	70.3dB	G#7-SineWave (-36dBFS)	73.9dB
G#7-Scream (-42dBFS)	63.3dB	G#7-SineWave (-42dBFS)	67.8dB
G#7-Scream (-48dBFS)	58.2dB	G#7-SineWave (-48dBFS)	62.2dB

Table 7.4

The final number of trials obtained for the loudness experiment is $20 \times 4 \times 3 \times 9 = 2160$ (*subjects* \times *timbres* \times *frequencies* \times *loudnesses*). The final number of trials for the frequency experiment is $20 \times 4 \times 22 = 1760$ (*subjects* \times *timbres* \times *frequencies*). Nevertheless, from the 2160 *loudness* and the 1760 *frequency* trials some will be removed because of noise and connection problems. The exact number of removed trials will be specified in the next chapter.

Chapter 8

Pre-Processing

The loudness and frequency experiments produced a set of EDF files saved from the TestBench. The EDF files were converted to CSV files and the converted files were used to pre-process the acquired signal. The following chapter addresses the implementation constructed for cleaning and preparing the signal and the different tools used to develop this implementation.

8.1 Ordering and Cutting the Trials

Each experiment produced one file per subject. From the 40 experiment files taken from 20 subjects, 2 loudness experiment files and 2 frequency experiment files were removed from the dataset. The signals in these files contained a very high level of noise due to connection problems with the headset, therefore, all relevant information was lost.

Each subject's file contains from 12 to 14 minutes of EEG signals with a sampling rate of 129¹ samples per second. An ordering/cutting program was created to extract the trials from the subject's files and save them as text files. Furthermore, the program organizes the target values in correspondence to the trials and saves them as a text file. The content of each file was ordered by subject and different files were created for each timbre. Additionally, the program removes the trials with low contact quality or markers. Note that the markers were inserted during the execution of the experiments to indicate blinks or connection problems.

Each trial corresponds to one specific sound. The total length of a sound is approximately five seconds and the ordering/cutting program was designed

¹Initially a sampling rate of 128 samples per second was selected in the TestBench program. Nevertheless, this selection generated 129 samples per second in the CSV files and therefore a sampling of 129 was used for the analysis.

to extract the signal contained in an specific interval within those five seconds. An interval of 3000 ms seconds was selected for the frequency trials, from 150ms to 3150ms, resulting in 387 samples per trial. An interval of 2000ms was selected for the loudness trials, from 0ms to 2000ms, resulting in 258 samples per trial. This selection of intervals was based on experience from previous research about the human perception of loudness and frequency.

A sensation of a frequency, measured in cycles per second or Hz, can be established about four to eight cycles after the onset of a sound and there is a certain level of recognition just from the onset. Nevertheless, certainty about the frequency of a sound requires a minimum duration. Achieving a finer frequency resolution requires a corresponding longer time interval [11]. An interval of three seconds was selected to allow a fine frequency resolution for all the experiment frequencies and the onset of the sound was removed to eliminate the original uncertain perception of frequency based on the onset.

The perception of loudness is biased for sounds shorter than 200ms. The ear averages over approximately 200ms to determine the loudness of a sound. For a shorter sound, loudness is perceived as proportional to the sound duration [11]. Therefore a two second interval was selected to ensure that the perceived loudness corresponded to the measured loudness of the sustain of the sound and cutting out the release of the different timbres.

A summary of the trials obtained from the cutting/ordering program is shown in Table 8.1. Note that the dataset size corresponds to the number of channels and the size of the trials ($channels \times (trials * samplesPerTrial)$).

Experiment	Trials	Duration	Interval	Samples Per Trial	Dataset Size
Frequency	1519	2000ms	0ms – 2000ms	387	14 × 587853
Loudness	1890	3000ms	150ms – 3150ms	258	14 × 487620

Table 8.1

8.1.1 Filtering

The loudness and frequency datasets were filtered to remove noise. The first filter applied to the datasets was a band-pass filter between 1 and 50Hz. This filter was implemented in MATLAB with EEGLab [72] using an elliptic infinite impulse response filter, also known as Cauer filter [73].

After the first band-pass filter, the datasets were filtered in order to separate the different brain rhythms. This separation was used to determine, during the data analysis, which rhythm contains the more relevant information about frequency/loudness processing and perception. The rhythms selected for this filtering process were Delta (1-4Hz), Theta (4-7Hz), Alfa (8-14Hz) and Beta (14-30Hz). Additionally, the interval from 1Hz to 50Hz was saved to determine if there is relevant information in the frequencies that were not included in the selection of rhythms.

8.1.2 Transformation of the Target Values

For the frequency experiment, the target values, originally measured in Hz were transformed from their exponential scale to a more natural scale. This transformation linearized the targets based on the human perception of tones, as exemplified in Equation 2.2.

The targets were transformed using the equation $y_{scaled} = 10 (\log_{10}(y_{original})) - 14$, this equation transforms the frequencies shown in Table 7.3 into a linear scale between 1 and 22, with each frequency separated by approximately 1.0.

Chapter 9

Data Analysis Implementation

This chapter will address the implementation of the data analysis phase of this project. Data analysis covers the transformation from the trials to Sample Covariance Matrices (SCMs) in Riemannian space and the projection of the SCMs into tangent space, where different regression learning methods can be applied. Afterwards, Ridge regression and Dual Representation Regression are used to find a linear model relating the SCMs in tangent space to their target frequency or loudness values.

9.1 Feature Transformation: Riemannian Space to Tangent Space

The implementation of the feature transformation process was constructed in MATLAB [74] using the Covariance Toolbox [75]. This process covers the transformation of all the trials into SCMs in Riemannian space and all the necessary calculations to project this matrices into tangent space, where the regression is computed. Algorithm 2 shows a summary of the Riemannian theory in Chapter 4 and the pseudocode of the feature transformation implementation.

Algorithm 2: Feature Transformation

Data: I trial matrices \mathbf{X}_i (Equation 3.1)

Result: I feature vectors $\text{vect}(\tilde{\mathbf{S}}_i)$

Calculate Sample Covariance Matrices;

forall the trials \mathbf{X}_i do

┌ $\mathbf{P}_i = \frac{1}{s-1} \mathbf{X}_i \mathbf{X}_i^T$ (Equation 3.3);

Calculate Geometric Mean \mathfrak{G} (Algorithm 1);

$\mathbf{P}_{ref} = \mathfrak{G};$

Feature Transformation ($\tilde{\mathbf{S}}_i = \phi(\mathbf{P}_i)$);

forall the \mathbf{P}_i do

┌ $\tilde{\mathbf{S}}_i = \log(\mathbf{P}_{ref}^{-1/2} \mathbf{P}_i \mathbf{P}_{ref}^{-1/2})$ (Equation 5.3 and 5.4);

Vectorization;

forall the $\tilde{\mathbf{S}}_i$ do

┌ $\tilde{\mathbf{S}}_i = \text{vect}(\tilde{\mathbf{S}}_i)$ (Equation 4.3);

9.2 Ridge Regression

The ridge regression implementation was constructed using MATLAB [74]. This process covers the steps necessary for the calculation of the weights for the features. The features are contained in the vectors \tilde{S}_i produced by the feature transformation. A summary of the regression theory explained in Section 5.3 and the pseudocode of this implementation is shown in Algorithm 3.

Algorithm 3: Ridge Regression

Data: I feature vectors \tilde{S}_i , N runtime feature vectors \tilde{S}_i^{run} , λ value, vector \mathbf{y} of I targets

Result: A vector \mathbf{y}_{est} of N estimations

Construct feature matrix Φ ;

$\Phi = \text{concatenateRows}(\tilde{S}_1, \tilde{S}_2, \dots, \tilde{S}_I)$;

Add column of 1's to Φ (Equation 5.6);

Construct feature matrix Φ_{run} ;

$\Phi_{run} = \text{concatenateRows}(\tilde{S}_1^{run}, \tilde{S}_2^{run}, \dots, \tilde{S}_N^{run})$;

Add column of 1's to Φ_{run} (Equation 5.6);

Calculate weights \mathbf{w} ;

$\mathbf{w} = (\Phi^T \Phi + \lambda I)^{-1} \Phi^T \mathbf{y}$ (Equation 5.18);

Calculate estimations;

$\mathbf{y}_{est} = \Phi_{run} \mathbf{w}$;

9.3 Dual Representation Regression with Riemannian-based kernel

The Dual Representation Regression with Riemannian-based kernel was implemented using MATLAB [74]. This process covers the steps necessary for the estimation of the target frequencies or loudness levels. These target values, and their corresponding trials, were separated from the data set in order to evaluate the regression model. Algorithm 4 shows a summary of the theory covered in Section 5.4 and the pseudocode of this implementation.

Algorithm 4: Dual Representation Regression with Riemannian-based kernel

Data: I feature vectors \tilde{S}_i , N runtime feature vectors \tilde{S}_i^{run} , λ value, vector \mathbf{y} of I targets

Result: A vector \mathbf{y}_{est} of N estimations.

Construct feature matrix Φ ;

$\Phi = \text{concatenateRows}(\tilde{S}_1, \tilde{S}_2, \dots, \tilde{S}_I)$;

Add column of 1's to Φ (Equation 5.6);

Construct feature matrix Φ_{run} ;

$\Phi_{run} = \text{concatenateRows}(\tilde{S}_1^{run}, \tilde{S}_2^{run}, \dots, \tilde{S}_N^{run})$;

Add column of 1's to Φ_{run} (Equation 5.6);

Construct Gram Matrix \mathbf{K} ;

$\mathbf{K} = \Phi\Phi^T$ (Equation 5.25);

Construct kernel \mathbf{k} ;

$\mathbf{k} = \Phi_{run}\Phi^T$;

Calculate estimations;

$\mathbf{y}_{est} = \mathbf{k}(\mathbf{K} + \lambda\text{Ident})^{-1}\mathbf{y}$ (Equation 5.28);

Chapter 10

Interpolation Implementation

The implementation of the Riemannian Interpolation process was constructed in MATLAB [74] using the Covariance Toolbox [75]. The process covers the steps necessary to estimate a covariance matrix given a *frequency* or *loudness* level between the minimum and maximum observed targets. Algorithm 5 shows a summary of the pseudocode of the Riemannian Interpolation explained in Chapter 6.

Algorithm 5: Riemannian Interpolation

Data: I Sample Covariance Matrices P_i , vector \mathbf{y} of I targets, y_r target

Result: SCM Estimation \hat{P}_r .

Get the two closest targets to y_r , $\forall y_i < y_r$ and $\forall y_j > y_r$, with $y_i, y_j \in \mathbf{y}$;

$y_1 = \text{minimumDistance}(y_i, y_r)$;

$y_2 = \text{minimumDistance}(y_r, y_j)$;

Compute the Geometric Means \mathfrak{G}_1 and \mathfrak{G}_2 , (Algorithm 1);

Calculate the \mathfrak{G}_1 of all P_i with target value y_1 ;

Calculate the \mathfrak{G}_2 os all P_j with target value y_2 ;

Set P_{mean_1} and P_{mean_2} as;

$P_{mean_1} = \mathfrak{G}_1$;

$P_{mean_2} = \mathfrak{G}_2$;

Calculate the distance ratio γ ;

$\gamma = \frac{y_r - y_1}{y_2 - y_1}$ (Equation 6.6);

Project the P_{mean} that is furthest away into the tangent space of the closest P_{mean} ;

$S_2 = \text{Log}_{P_{mean_1}}(P_{mean_2})$ (Equation 6.4);

Estimate of the P_r matrix;

$\hat{P}_r = \text{Exp}_{P_{mean_1}}(\gamma S_2)$ (Equation 6.5);

Part IV

Results and Conclusion

Chapter 11

Results

The feature transformation and regression implementations were tested to determine the validity of the model generated from the brain's response to single sound-parameter variation.

The following tests were executed using the Ridge Regression prediction method exclusively. Note that for this project's case, the Dual Representation Regression is equivalent to the Ridge Regression because of the use of an explicit feature transformation.

Different subsets of channels and different frequency bands were tested in order to improve the results of the regression and cross validation models. Table 11.1 shows a summary of the frequency bands and the channels' subset size used for this model selection.

Channels' Subset Size	Frequency Band
14 channels	1-50Hz
8 channels	1-4Hz
6 channels	4-7Hz
1 channel	8-14Hz
	13-30Hz

Table 11.1

To improve the results, all combinations of channels' subsets sizes and frequency bands shown in Table 11.1 were tested in order to determine what channels and frequency bands contained more relevant information for the regression model. The model selection was executed using cross validation

results as the selection criteria.

11.1 Visual Interpretation of the Results

The results presented throughout this chapter are illustrated with different graphs. These graphs show a visualization of the results obtained from the regression and cross validation methods applied to the different datasets. Coordinate systems with three different combinations of axes are used, trials against targets, targets against estimations and targets against residuals.

The first combination, trials against targets, can only be used for the datasets containing one channel. The calculation of the SCMs for one channel trials produces a scalar number per trial. This scalar value in Riemannian space also corresponds to a scalar value in tangent space. The scalar value in tangent space of each trial is plotted in the x-axis of the coordinate system and its corresponding target is plotted in the y-axis. In this coordinate system, the regression function \mathbf{w} can also be plotted, for scalar trials the regression is a linear function of the form $w_1x + w_0$.

For the analysis of datasets containing trails with multiple channels, the preceding combination cannot be visualized in two dimensions because the trials are vectors or multi-dimensional points. Therefore, the validity of the regression model is shown by plotting the target values in the x-axis, against their corresponding estimation in the y-axis. An accurate regression would create estimation values that are close to their corresponding target. Therefore, the accuracy of the regression function is proportional to the distance between the points and the diagonal created by the function $y = x$.

The third axes combination can be used for any number of channels and it shows how well each target value was estimated by the regression function. The target values are plotted in the x-axis and the residuals in the y-axis. An accurate regression would create y-values close to zero.

11.2 Cross Validation

Cross validation is a validation method that determines how well a model generalizes to external data sets. For this project, it evaluates how well the regression model predicts unseen values.

A k -fold cross validation method divides the dataset into k subsets. One of the subsets is excluded from the dataset and the remaining $k - 1$ subsets

are used to compute the regression model. Afterwards, the excluded subset is used to evaluate the regression model using the mean squared residual (MSR).

$$\text{MSR} = \frac{1}{N} \sum_{i=1}^N (y_i - y_i^{est})^2 \quad (11.1)$$

where y_i is the target value and y_i^{est} is the estimated value for the i -th trial of the excluded subset.

This k -fold separation process is repeated k times excluding a different subset in each iteration. After this process is finished an average error is calculated using the MSR of every iteration.

$$E_{\text{avg}} = \frac{1}{k} \sum_{i=1}^k \text{MSR}_i \quad (11.2)$$

where k is the number of subsets. This averaged error represents an estimation of the regression model's quality. The functionality and implementation of the k -fold cross validation method is summarized in Algorithm 6.

Algorithm 6: k -Fold Cross Validation

Data: Dataset of I trials, vector \mathbf{y} of I targets, number of folds k

Result: Average error E_{avg}

split_folds = split(Dataset into k folds);

split_targets = split(\mathbf{y} into k subsets);

forall the f in *split_folds* and t in *split_targets* **do**

- 1 $\left| \right.$ $\text{runtime_trials} = f$;
- 2 $\left| \right.$ $\mathbf{y}_{\text{runtime}} = t$;
- 3 $\left| \right.$ $\text{training_trials} = (k - 1)$ folds, excluding f ;
- 4 $\left| \right.$ $\mathbf{y}_{\text{training}} = (k - 1)$ target subsets, excluding t ;
- 5 $\left| \right.$ Calculate the weight vector \mathbf{w} using the *training_trials* and $\mathbf{y}_{\text{training}}$;
- 6 $\left| \right.$ Calculate the $\mathbf{y}_{\text{estimations}}$ for the *runtime_trials* using \mathbf{w} ;
- 7 $\left| \right.$ Calculate MSR between $\mathbf{y}_{\text{runtime}}$ and $\mathbf{y}_{\text{estimations}}$ (Equation 11.1);

Calculate the error E_{avg} of all the MSRs (Equation 11.2);

Leave One Subject Out Cross Validation (LOSO)

One of the k -fold cross validation methods applied in this project is the Leave One Subject Out (LOSO) cross validation. This cross validation method was used to evaluate the regression model created for the complete dataset. This dataset contains all the subjects and all the timbres.

For the frequency experiment, the size of each fold is between 85 and 88, depending on the amount of trials removed because of noise and artifacts during the pre-processing phase. For the loudness experiment, the size of each fold is between 100 and 108 depending on the amount of trials removed in the pre-processing phase.

Leave One Trial Out Cross Validation (LOTO)

Another k -fold cross validation method used in this project is the Leave One Trial Out (LOTO) cross validation. This cross validation method was used to evaluate the regression model created for each subject.

LOTO cross validation is accomplished by using Algorithm 6, with the number of folds equal to the number of trials ($k = I$) and consequently making the size of each fold equal to 1.

Leave One Timbre Out Cross Validation (LOTiO)

A k -fold cross validation method that is very relevant for this project is the Leave One Timbre Out (LOTiO) cross validation. Each experiment was designed with four different timbres and these timbres were selected to produce different sensations in the subject. Therefore, based on the results, assumptions could be made about the subject's reaction to the different timbres. In other words, we could either observe a general reaction to loudness/frequency changes for all the timbres or we could observe that the reaction is timbre dependent.

In this k -fold cross validation method the size of each fold is approximately 25% of the trials, making $k = 4$. This cross validation method was used to evaluate the regression model created for the dataset of all the subjects and the regression model created for each subject.

11.3 Loudness Experiment - All Subjects

This section presents the results obtained from the loudness experiment using the dataset of all the subjects. First, the results of the analysis using 14 channels and all the timbres is shown. Afterwards, the analysis using 1 channel and all timbres is presented. Finally, the analysis using 14 channels and separate timbres is discussed.

11.3.1 LOSO Cross Validation, 14 Channels, All Timbres

LOSO cross validation using the set of all 14 channels was applied to the dataset containing all the subjects and timbres. Figure 11.1 shows the target values of the training set ($\mathbf{y}_{training}$) against their corresponding estimation in the first LOSO iteration.

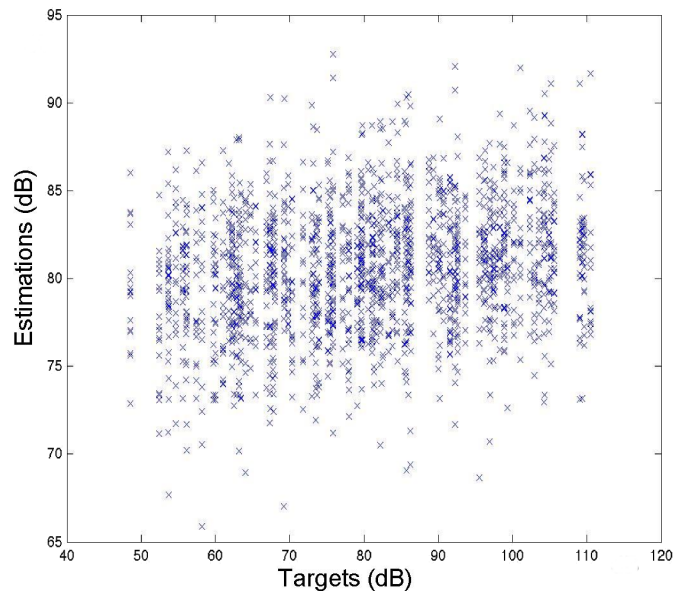


Figure 11.1: Loudness Experiment - Training targets $\mathbf{y}_{training}$ against their corresponding estimation. The regression function was estimated using the trials of subjects 2-18 with all 4 timbres and a frequency band of 1-50Hz.

The estimations shown in Figure 11.1, demonstrate that the calculated regression function is unable to predict the target values of the training set ($\mathbf{y}_{training}$). We chose to plot the estimations for the training targets ($\mathbf{y}_{training}$) instead of the runtime targets ($\mathbf{y}_{runtime}$) in order to show the behavior of majority of the dataset. Note that the estimations of the training set and runtime set show analogous behaviors. LOSO cross validation with

14 channels and all four timbres produces similar results for all the different frequency bands.

The inaccuracy of the regression function can be induced by one or more of the following reasons:

Low Spatial Resolution Not enough spatial information about the brain is obtained by the headset. In other words, the restricted number of channels may cause the information to be concentrated in specific regions of the brain. If relevant information about the relationship between loudness and the brain's state is in other regions, the headset may not be able to read the necessary information.

Nonlinear Relationship between Loudness and the Brains's State

If the relationship between loudness and the state of the brain is nonlinear in Riemannian space, further feature transformation may be required to find a linear relationship. One could imagine that applying another kernel to transform the features into another space, as explained in Section 5.29, might yield better results.

Loudness to Brain's State Relationship is Timbre Dependent Each timbre may cause different effects in the brain's state. This would render the regression model unable to find a linear relationship between the loudness and the brain's state if more than one timbre is involved.

Loudness to Brain's State Relationship is Subject Dependent Each subject has a different perception of loudness that may induce different reactions in the brain. This would render the regression model unable to find a linear relationship for more than one subject at a time.

11.3.2 LOSO Cross Validation, 1 Channel, All Timbres

In order to evaluate if any of the channels contained relevant information for the regression model, LOSO cross validation using all timbres was executed for every channel. Furthermore, a coordinate system showing trials against targets, as explained in Subsection 11.1, was used to visually inspect each channel's regression. Figure 11.2 shows an example of the obtained results. The plot shows the training trials for the first iteration of the LOSO cross validation using Channel FC6.

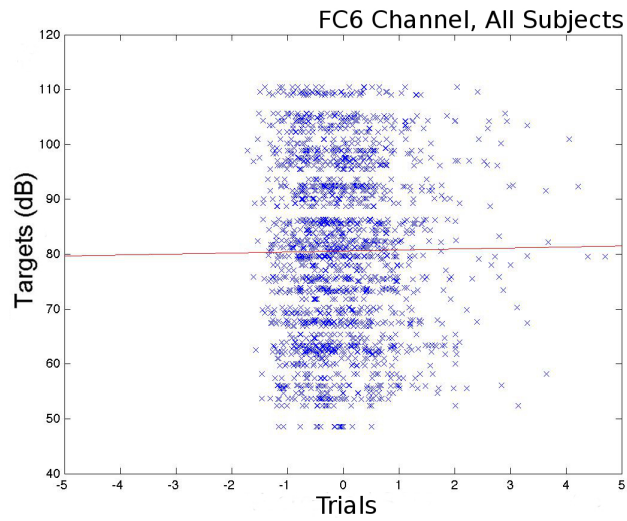


Figure 11.2: Loudness Experiment - Training trials against training targets ($y_{training}$) for Channel FC6. The regression function, depicted as a red line, was estimated using the trials of subjects 2-18 with all 4 timbres and a frequency band of 1-50Hz.

From Figure 11.2 we can observe that the trials are scattered in a pattern that does not show a linear behavior. Given the nonlinear distribution of the trials the constructed regression function is unsatisfactory. Figure 11.3 shows the estimations by target and the residuals by target that correspond to Figure 11.2.

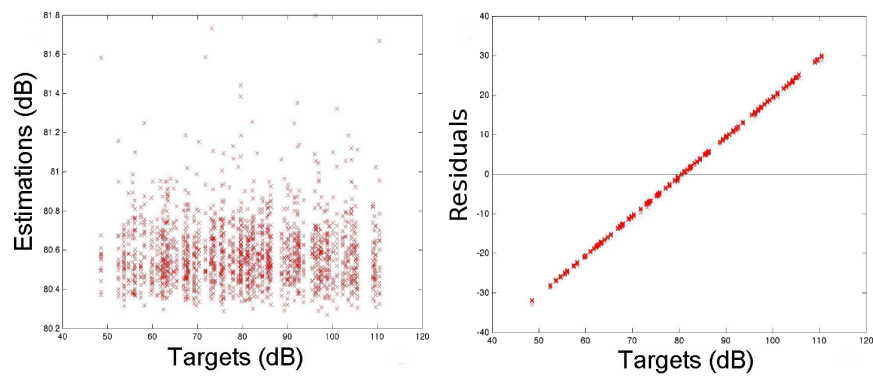


Figure 11.3: Loudness Experiment - Left plot shows training targets against training estimations. Right plot shows training targets against training residuals. Both plots correspond to the regression function shown in Figure 11.2.

All other channels and all frequency bands show results analogous to the previous example. After running all the combinations of channels and frequency bands we concluded that the signal contained in one single channel does not provide enough information to create a linear regression.

11.3.3 LOSO Cross Validation, 14 Channels, Separate Timbres

All previous attempts at finding a linear regression use the dataset containing all subjects and all timbres. For each timbre, a LOSO cross validation using 14 channels was computed in order to verify if the inaccuracy of the previous regressions was induced by combining all four different timbres. Figure 11.4 shows the estimations of the training set for the first iteration of the LOSO cross validation of each timbre.

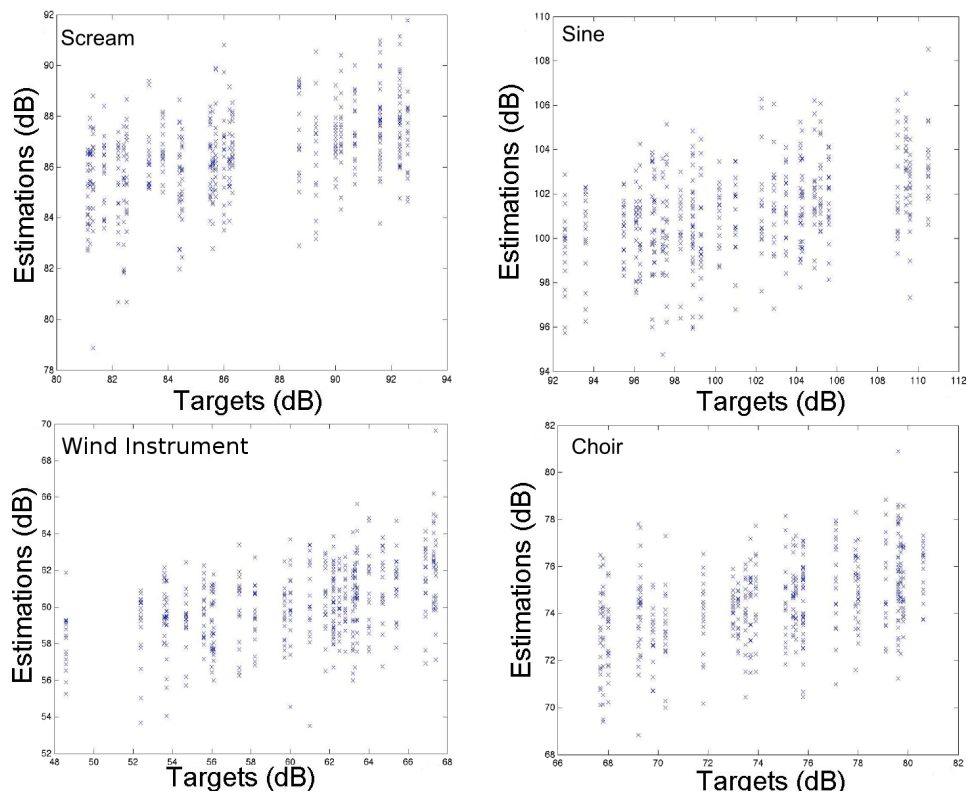


Figure 11.4: Loudness Experiment - Training targets $y_{training}$ against their corresponding estimation. Separate timbres. All regression estimations were calculated using the trials of subjects 2-18 with a frequency band of 1-50Hz.

From Figure 11.4 we can infer that combining timbres was not the cause

for the inaccuracy of the previous regression functions. Each separate timbre shows different estimations that do not suggest a linear behavior for all subjects.

11.4 Loudness Experiment - Separate Subjects

For every subject, it is possible to find a representative linear regression model relating loudness level and the brain's state. This section presents the results obtained from the dataset separated by subject. First, the results of the LOTiO cross validation analysis using 14 channels is shown. Afterwards, LOTO cross validation is applied to different sets of channel. Throughout this section, cross validation error is used as the selection criterion for the model selection executed to improve the accuracy of the regression.

All the above mentioned cross validation analyses were executed for all frequency bands. The frequency band with the smallest cross validation error for all the different analyses was the θ rhythm (4-7Hz), shown in Table 3.2. Note that this is consistent with the research studies shown in Related Works, Section 2.5. Therefore, all the following cross validation analyses are applied to the θ rhythm filtered dataset.

11.4.1 LOTiO Cross Validation, 14 Channels

After determining that a linear relationship could be found for the training set of all the subjects, LOTiO was applied in order to observe if the relationship between loudness and the brain's state is timbre independent. Figure 11.5 shows the results of the LOTiO cross validation for Subject 16, first iteration.

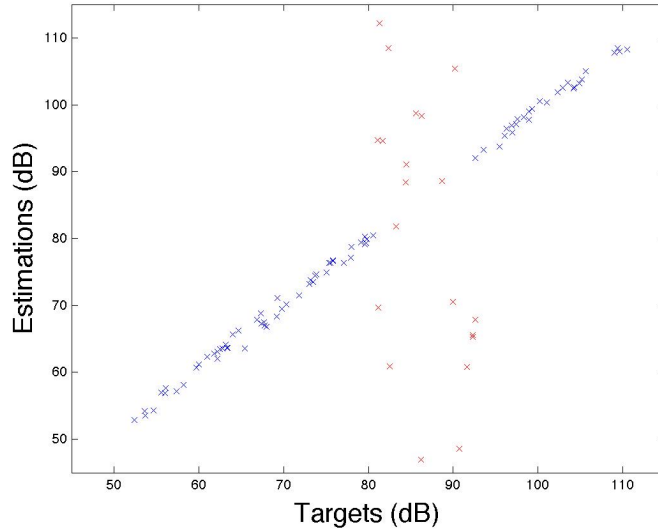


Figure 11.5: Loudness Experiment - Training Targets $\mathbf{y}_{training}$ against their corresponding estimation in blue. Runtime targets $\mathbf{y}_{runtime}$ against their corresponding estimation in red. The regression function was estimated for Subject 16 using the timbres Choir, Scream and Sine and a frequency band of 4-7Hz. The runtime estimations were calculated using the Winds' trials.

Figure 11.5 shows that the regression function estimates the training targets very accurately, but estimates the runtime targets loosely. All four iterations of the LOTiO cross validation show similar results. The inability of the regression function to predict the runtime set accurately may be caused by overfitting or because the relationship between loudness and the brain's state is timbre dependent. In order to determine if the regression function is overfitting to the training set, LOTO cross validation is applied to each subject instead of LOTiO cross validation.

11.4.2 LOTO Cross Validation, 14 Channels, All Timbres

LOTO cross validation using the set of all 14 channels was applied to the dataset of each subject, using all four timbres. Figure 11.6 shows an example of the estimations of the training targets and the estimation of the runtime target of every iteration.

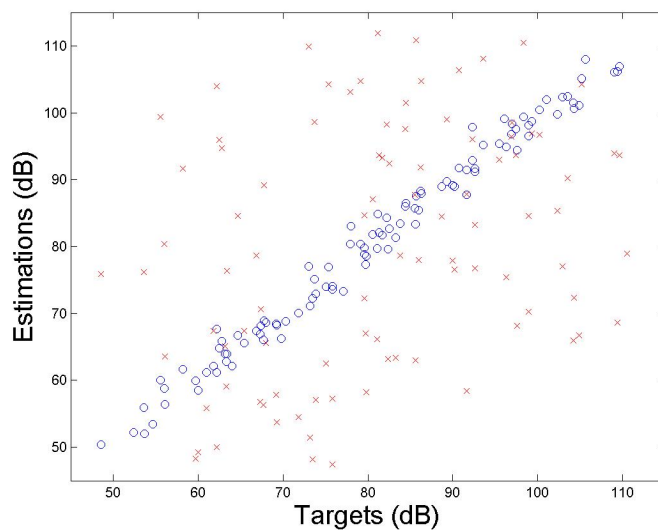


Figure 11.6: Loudness Experiment - Blue Circles show the training targets ($\mathbf{y}_{\text{training}}$) against their corresponding estimation, for the last LOTO iteration. Red crosses show the estimation of the runtime target for all LOTO iterations. The regression function of every iteration was calculated using Subject 8, $\lambda = 1$ and a frequency band of 4-7Hz.

The LOTO cross validation for Subject 8 has a cross validation error (E_{avg}) of 476.76. Table 11.2 shows the error E_{avg} of each subject. The average cross validation error for all 18 subjects is 787.08.

Subject	Cross Validation Error	Subject	Cross Validation Error
1	1385.80	10	754.44
2	676.04	11	1618.00
3	196.45	12	1287.90
4	796.10	13	556.41
5	1013.50	14	337.52
6	1364.50	15	493.73
7	648.43	16	323.68
8	476.76	17	1137.30
9	383.73	18	717.16

Table 11.2

The results obtained for the LOTO cross validation suggest that the regression function of every iteration overfits on the training set causing the

runtime estimation to be inaccurate, as explained in Section 5.2. Furthermore, the results obtained for the LOTO cross validation are similar to the results of LOTiO cross validation. This suggests that the error (E_{avg}) in LOTiO cross validation is not induced by the timbre dependence but by overfitting.

The complexity penalty parameter (λ) of the regression model can be tuned in order to balance the overfitting and underfitting of the model. Increasing the value of λ causes the regression to decrease the number of features used. Further experimentation revealed that $\lambda = 15$ decreases the E_{avg} significantly and higher λ values do not produce a significant decrease. Figure 11.7 shows the result of setting $\lambda = 15$ for the LOTO cross validation shown in Figure 11.6.

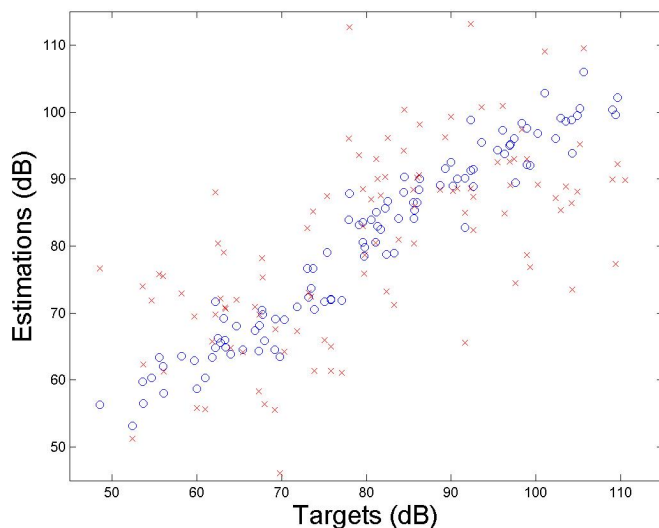


Figure 11.7: Loudness Experiment - Blue Circles show the training targets ($\mathbf{y}_{\text{training}}$) against their corresponding estimation, for the last LOTO iteration. Red crosses show the estimation of the runtime target for all LOTO iterations. The regression function of every iteration was calculated using Subject 8, $\lambda = 15$ and a frequency band of 4-7Hz.

The LOTO cross validation for Subject 8, using $\lambda = 15$, has a cross validation error (E_{avg}) of 168.48. Table shows the error (E_{avg}) of each subject. The average cross validation error for all 18 subjects is 285.69.

Subject	Cross Validation Error	Subject	Cross Validation Error
1	399.75	10	282.92
2	249.29	11	453.59
3	144.48	12	402.92
4	275.02	13	233.21
5	389.12	14	163.46
6	421.62	15	207.40
7	379.84	16	118.67
8	168.48	17	376.95
9	148.33	18	327.45

Table 11.3

As shown in Table 11.3, tuning the λ parameter achieves a significant improvement. Nevertheless, other approaches that decrease the number of features/predictors for the regression model may yield better results.

11.4.3 LOTO Cross Validation, 8 Channels, All Timbres

The number of predictors, for the regression function, generated by using all 14 channels is 106. The number of trials per subject is in average 104. Therefore, a contributing factor to the overfitting of the regression function is that the number of predictors surpasses the number of observations.

Reducing the number of channels decreases the number of predictors for the regression function significantly. Model selection with cross validation error as the selection criterion was used to determine which pairs of channels contain the most relevant information for the regression model. The ranking of each pair of channels was used to select the subset of eight channels that yields the best results. Figure 11.8 shows the scalp map for the eight selected channels.

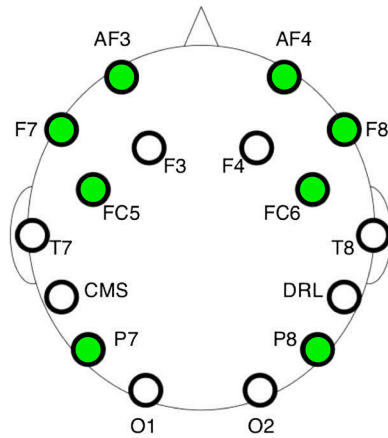


Figure 11.8: Subset of eight channels selected for the loudness experiment.

LOTO cross validation using the set of channels shown in Figure 11.8 was applied to each subject, using all four timbres. Figure 11.9 shows an example of the estimations of the training targets and the estimation of the runtime target of every iteration.

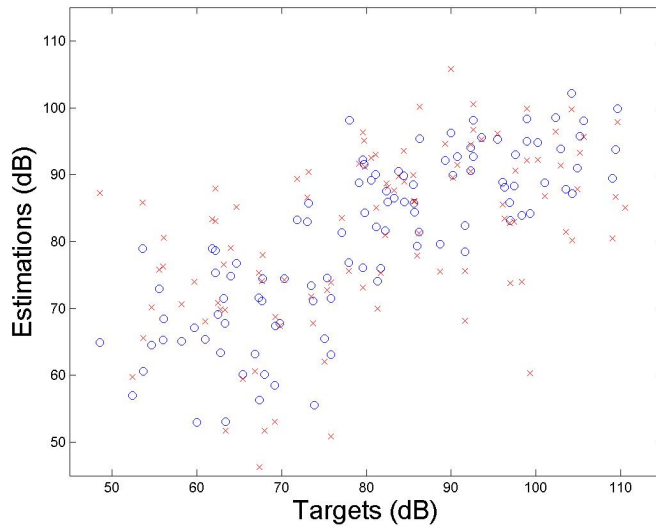


Figure 11.9: Loudness Experiment - Blue Circles show the training targets ($\mathbf{y}_{\text{training}}$) against their corresponding estimation, for the last LOTO iteration. Red crosses show the estimation of the runtime target for all LOTO iterations. The regression function of every iteration was calculated using Subject 8, $\lambda = 1$ and a frequency band of 4-7Hz.

The LOTO cross validation for Subject 8, using eight channels, has a cross validation error (E_{avg}) of 227.34. Table 11.4 shows the error (E_{avg}) of each subject. The average cross validation error for all 18 subjects is 267.61.

Subject	Cross Validation Error	Subject	Cross Validation Error
1	304.93	10	307.64
2	259.45	11	313.84
3	184.82	12	341.64
4	208.95	13	263.42
5	315.00	14	120.43
6	370.78	15	245.88
7	371.76	16	135.97
8	227.34	17	328.49
9	156.07	18	360.54

Table 11.4

As shown for the 14 channel LOTO cross validation, tuning the complexity penalty parameter of the regression improves the accuracy of the model significantly. Figure 11.10 shows the result of setting $\lambda = 15$ for the LOTO cross validation shown in Figure 11.9.

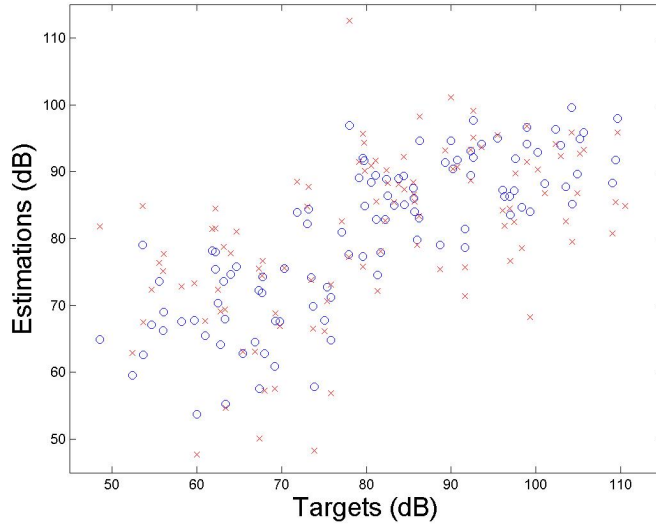


Figure 11.10: Loudness Experiment - Blue Circles show the training targets ($\mathbf{y}_{training}$) against their corresponding estimation, for the last LOTO iteration. Red crosses show the estimation of the runtime target for all the LOTO iterations. The regression function of every iteration was calculated using Subject 8, $\lambda = 15$ and a frequency band of 4-7Hz.

The LOTO cross validation for Subject 8, using eight channels and $\lambda = 15$, has a cross validation error (E_{avg}) of 190.04. Table shows the error (E_{avg}) of each subject. The average cross validation error for all 18 subjects is 227.57.

Subject	Cross Validation Error	Subject	Cross Validation Error
1	260.09	10	250.98
2	216.18	11	263.74
3	156.22	12	289.55
4	187.14	13	224.47
5	268.32	14	104.05
6	319.78	15	214.77
7	314.78	16	119.89
8	190.04	17	273.07
9	133.70	18	309.43

Table 11.5

11.4.4 LOTO Cross Validation, 6 Channels, All Timbres

To further reduce the number of predictors for the regression model, the three pairs of channels that contained the most relevant information were selected. Figure 11.11 shows the scalp map for the six selected channels.

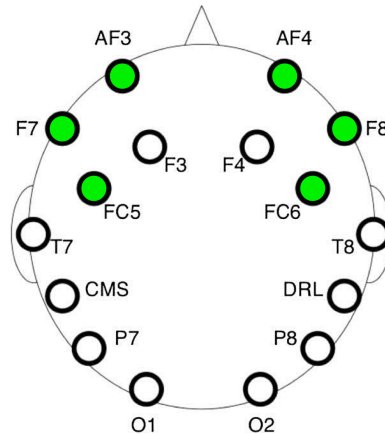


Figure 11.11: Subset of six channels selected for the loudness experiment.

LOTO cross validation using the set of channels shown in Figure 11.11 was applied to each subject, using all four timbres. Figure 11.12 shows an example of the estimations of the training targets and the estimation of the runtime target for every iteration.

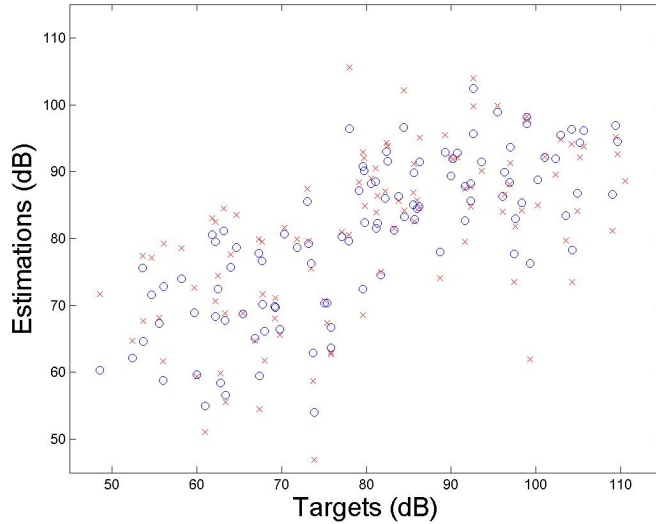


Figure 11.12: Loudness Experiment - Blue Circles show the training targets ($\mathbf{y}_{\text{training}}$) against their corresponding estimation, for the last LOTO iteration. Red crosses show the estimation of the runtime target for all LOTO iterations. The regression function of every iteration was calculated using Subject 8, $\lambda = 1$ and a frequency band of 4-7Hz.

The LOTO cross validation for Subject 8, using six channels, has a cross validation error (E_{avg}) of 141.48. Table 11.6 shows the error (E_{avg}) of each subject. The average cross validation error for all 18 subjects is 239.60.

Subject	Cross Validation Error	Subject	Cross Validation Error
1	272.50	10	278.00
2	199.62	11	276.91
3	213.42	12	341.64
4	227.22	13	263.42
5	279.37	14	255.60
6	320.10	15	134.198
7	173.19	16	209.77
8	141.48	17	286.05
9	272.98	18	256.78

Table 11.6

As shown for the 14 channels and eight channels LOTO cross validation,

tuning the complexity penalty parameter of the regression improves the accuracy of the model significantly. Figure 11.13 shows the result of setting $\lambda = 15$ for the LOTO cross validation shown in Figure 11.12.

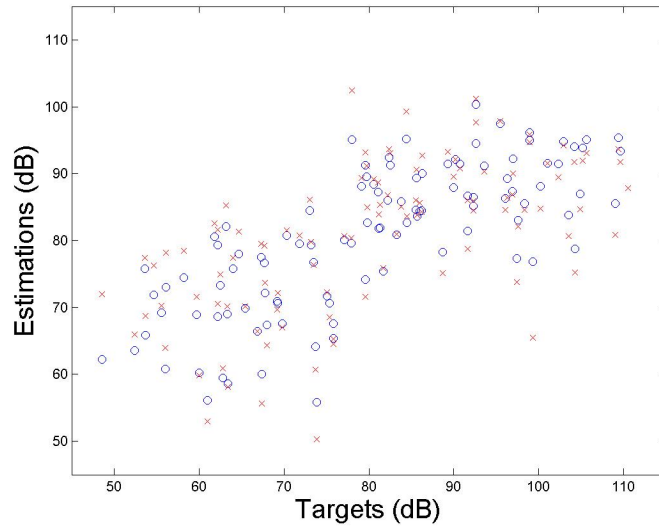


Figure 11.13: Loudness Experiment - Blue Circles show the training targets ($\mathbf{y}_{\text{training}}$) against their corresponding estimation, for the last LOTO iteration. Red crosses show the estimation of the runtime target for all the LOTO iterations. The regression function of every iteration was calculated using Subject 8, $\lambda = 15$ and a frequency band of 4-7Hz.

The LOTO cross validation for Subject 8, using six channels and $\lambda = 15$, has a cross validation error (E_{avg}) of 161.54. Table shows the error E_{avg} of each subject. The average cross validation error for all 18 subjects is 221.59.

Subject	Cross Validation Error	Subject	Cross Validation Error
1	255.42	10	248.15
2	183.81	11	259.41
3	188.21	12	259.25
4	213.34	13	231.85
5	259.16	14	125.02
6	297.39	15	193.13
7	287.78	16	194.27
8	161.54	17	255.85
9	131.70	18	243.43

Table 11.7

11.5 Separate Subjects - Summary

A summary of the results obtained for the loudness experiment using separate subjects is shown in Table 11.8.

λ	Channels' Subset Size	Average Cross Validation Error
1	14	787.08
15	14	285.69
1	8	267.61
15	8	227.57
1	6	239.60
15	6	221.59

Table 11.8

11.6 Frequency Experiment - All Subjects

This section presents the results obtained from the frequency experiment using the dataset of all the subjects. First, the results of the analysis using 14 channels and all the timbres is shown. Afterwards, the analysis using 1 channel and all timbres is presented. Finally, the analysis using 14 channels and separate timbres is discussed.

11.6.1 LOSO Cross Validation, 14 Channels, All Timbres

LOSO cross validation using the set of all 14 channels was applied to the dataset containing all the subjects and timbres. Figure 11.14 shows the target values of the training set ($\mathbf{y}_{training}$) against their corresponding estimation in the first LOSO iteration.

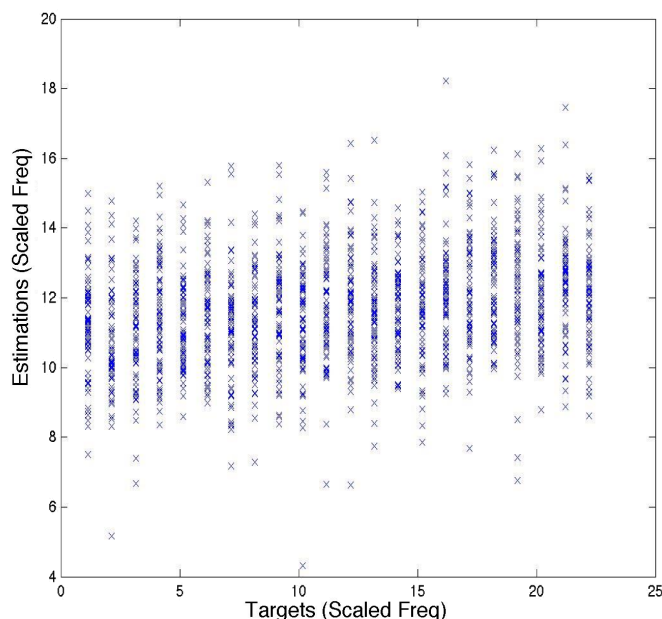


Figure 11.14: *Frequency Experiment - Training targets $\mathbf{y}_{training}$ against their corresponding estimation. The regression function was estimated using the trials of subjects 2-18 with all 4 timbres and a frequency band of 1-50Hz.*

The estimations shown in Figure 11.14, demonstrate that the calculated regression function is unable to predict the target values of the training set ($\mathbf{y}_{training}$). Analogous to the loudness experiment, we chose to plot the estimations for the training targets ($\mathbf{y}_{training}$) instead of the runtime targets ($\mathbf{y}_{runtime}$) in order to show the behaviour of majority of the dataset. The

estimations of the training set and runtime set show analogous behaviors. LOSO cross validation with 14 channels and all four timbres produces similar results for all the different frequency bands.

The inaccuracy of the regression function may be induced by similar reasons to the ones explained for the loudness experiment in Section 11.3. One or more of the the following reasons may cause the regression to be inaccurate:

- Low Spatial Resolution
- Nonlinear Relationship Between Frequency and the Brain's State
- Timbre Dependent Relationship Between the Frequency and the Brain's State
- Subject Dependent Relationship Between the Frequency and the Brain's State.

11.6.2 LOSO Cross Validation, 1 Channel, All Timbres

In order to evaluate if any of the channels contained relevant information for the regression model, LOSO cross validation using all timbres was executed for every channel. Furthermore, a coordinate system showing trials against targets, as explained in Subsection 11.1, was used to visually inspect each channel's regression. Figure 11.15 shows an example of the obtained results. The plot shows the training trials for the first iteration of the LOSO cross validation using Channel F7.

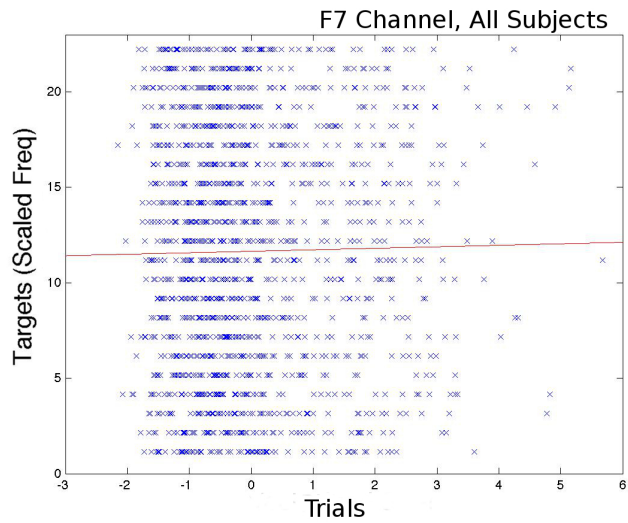


Figure 11.15: Frequency Experiment - Training trials against training targets (y_{training}) for Channel F7. The regression function, depicted as a red line, was estimated using the trials of subjects 2-18 with all 4 timbres and a frequency band of 1-50Hz.

From Figure 11.15 we can observe that the trials are scattered in a pattern that does not show a linear behavior. Given the nonlinear distribution of the trials the constructed regression function is unsatisfactory. Figure 11.16 shows the estimations by target and the residuals by target that correspond to Figure 11.15.

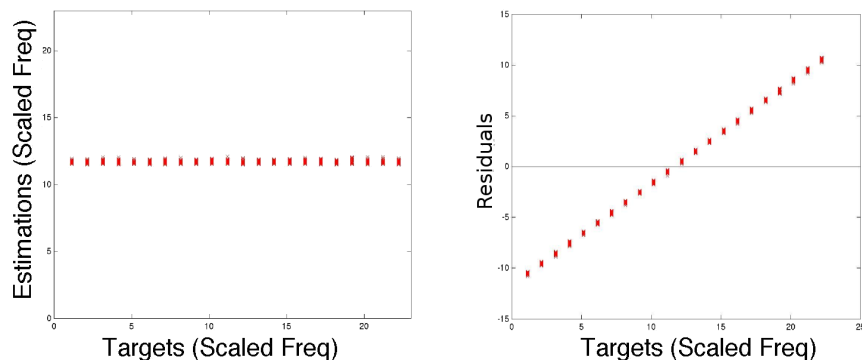


Figure 11.16: Frequency Experiment - Left plot shows training targets against training estimations. Right plot shows training targets against training residuals. Both plots correspond to the regression function shown in Figure 11.15.

All other channels and all frequency bands show results analogous to the previous example. After running all the combinations of channels and frequency bands we concluded that the signal contained in one single channel does not provide enough information to create a linear regression.

11.6.3 LOSO Cross Validation, 14 Channels, Separate Timbres

Following the same procedure as the loudness experiment, LOSO cross validation using 14 channels was computed in order to verify if the inaccuracy of the previous regressions was induced by combining all four different timbres. Figure 11.17 shows the estimations of the training set for the first iteration of the LOSO cross validation of each timbre.

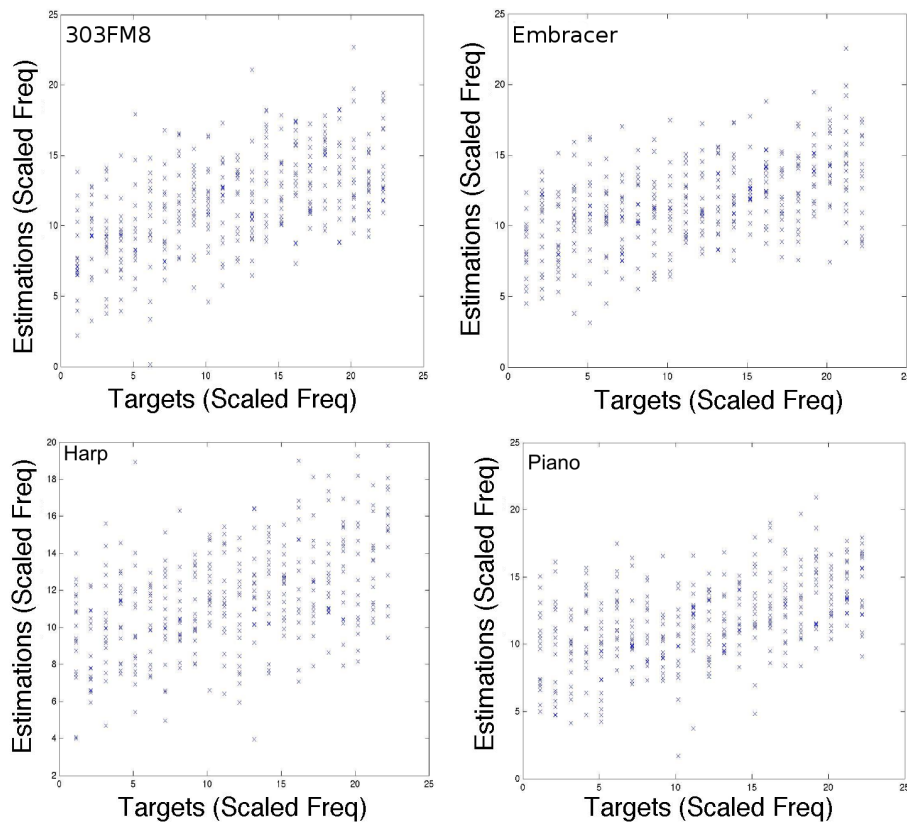


Figure 11.17: Frequency Experiment - Training targets y_{training} against their corresponding estimation. Separate timbres. All regression estimations were calculated using the trials of subjects 2-18 with a frequency band of 1-50Hz.

From Figure 11.17 we can infer that combining timbres was not the cause for the inaccuracy of the previous regression functions. Each separate timbre shows different estimations that do not suggest a linear behavior for all subjects.

11.7 Frequency Experiment - Separate Subjects

Coherently, on what has been found in the loudness experiment, it is possible to find a representative linear regression model relating frequency and the brain's state of single subjects. This section presents the results obtained from the dataset separated by subject. First, the results of the LOTiO cross validation analysis using 14 channels is shown. Afterwards, LOTO cross validation is applied to different sets of channels. Throughout this section, cross validation error is used as the selection criterion for the model selection executed to improve the accuracy of the regression.

All the above mentioned cross validation analyses were executed for all frequency bands. Similarly to the loudness experiment, the frequency band with the smallest cross validation error for all the different analyses was the θ rhythm (4-7Hz), shown in Table 3.2. Note that this is consistent with the research studies shown in Related Works, Section 2.5. Therefore, all the following cross validation analyses are applied to the θ rhythm filtered dataset.

11.7.1 LOTiO Cross Validation, 14 Channels

After determining that a linear relationship could be found for the training set of all the subjects, LOTiO was applied in order to observe if the relationship between loudness and the brain's state is timbre independent. Figure 11.18 shows the results of the LOTiO cross validation for Subject 7, first iteration.

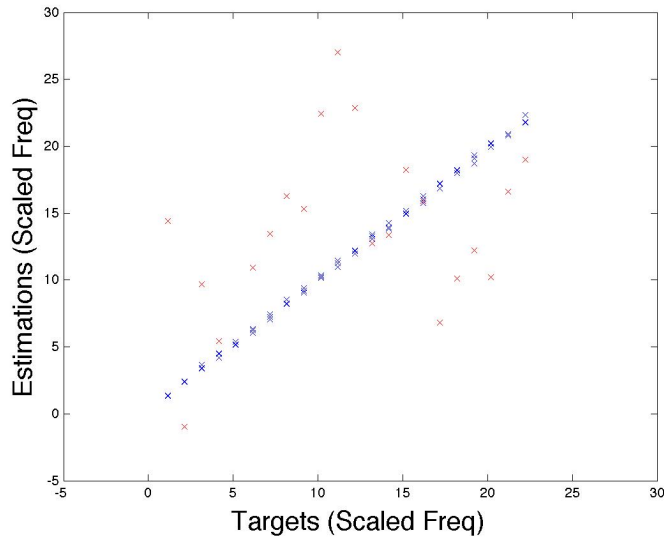


Figure 11.18: *Frequency Experiment - Training Targets $y_{training}$ against their corresponding estimation in blue. Runtime targets $y_{runtime}$ against their corresponding estimation in red. The regression function was estimated for Subject 7 using the timbres Embracer, Harp and Piano and a frequency band of 4-7Hz. The runtime estimations were calculated using the 303FM8's trials.*

Figure 11.18 shows that the regression function estimates the training targets very accurately, but estimates the runtime targets loosely. All four iterations of the LOTiO cross validation shows similar results. Analogous to the loudness experiment, in order to determine if the regression function is overfitting to the training set LOTO cross validation is applied to each subject instead of LOTiO cross validation.

11.7.2 LOTO Cross Validation, 14 Channels, All Timbres

LOTO cross validation using the set of all 14 channels was applied to the dataset of each subject, using all four timbres. Figure 11.19 shows an example of the estimations of the training targets and the estimation of the runtime target of every iteration.

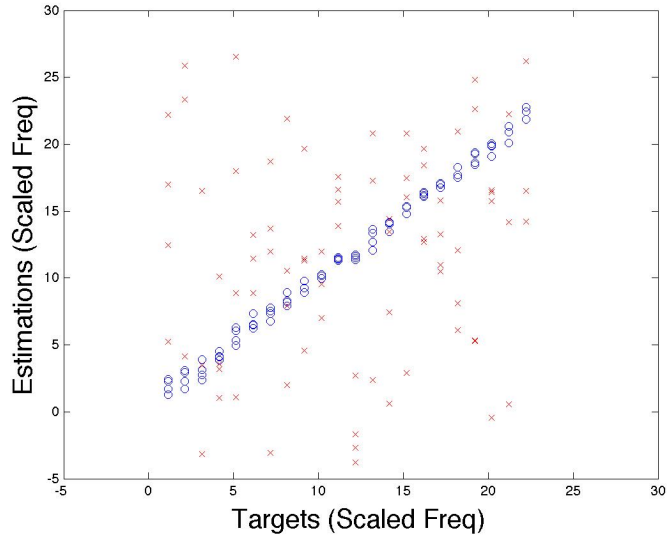


Figure 11.19: Frequency Experiment - Blue Circles show the training targets ($\mathbf{y}_{\text{training}}$) against their corresponding estimation, for the last LOTO iteration. Red crosses show the estimation of the runtime target for all LOTO iterations. The regression function of every iteration was calculated using Subject 1, $\lambda = 1$ and a frequency band of 4-7Hz.

The LOTO cross validation for Subject 1 has a cross validation error (E_{avg}) of 101.81. Table 11.9 shows the error E_{avg} of each subject. The average cross validation error for all 18 subjects is 127.12.

Subject	Average Model Error	Subject	Average Model Error
1	101.81	10	127.68
2	141.67	11	97.379
3	82.64	12	115.61
4	178.91	13	69.27
5	96.37	14	105.59
6	68.16	15	106.38
7	89.63	16	239.16
8	275.45	17	136.21
9	87.63	18	168.56

Table 11.9

Akin to the results obtained for the loudness experiments, the LOTO cross validation results show that the regression function overfits to the run-

time set. For this experiment the LOTO and LOTiO cross validation results are also similar, signifying that the error in LOTiO cross validation is mainly due to overfitting and not timbre dependence.

Experimentation reveals that tuning the complexity penalty parameter, also yields a significant improvement in the frequency experiment. Values below 15 achieve important improvements while values above 15 do not produce a significant gain. Figure 11.20 shows the results for the LOTO cross validation shown in Figure 11.19 using $\lambda = 15$.

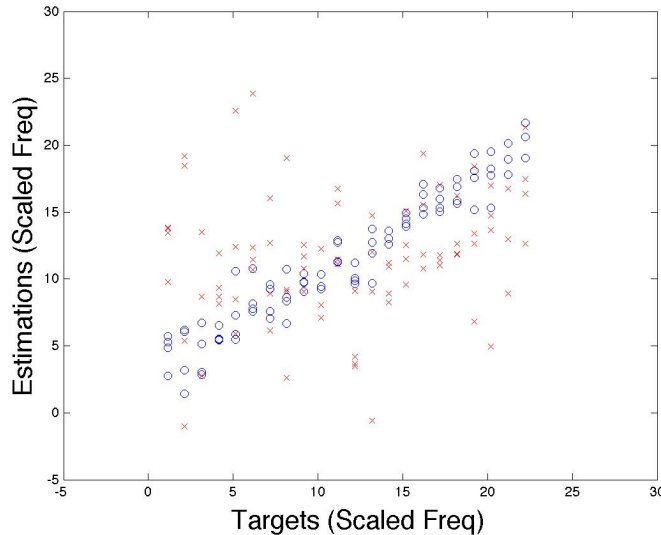


Figure 11.20: *Frequency Experiment - Blue Circles show the training targets ($\mathbf{y}_{training}$) against their corresponding estimation, for the last LOTO iteration. Red crosses show the estimation of the runtime target for all LOTO iterations. The regression function of every iteration was calculated using Subject 1, $\lambda = 15$ and a frequency band of 4-7Hz.*

The LOTO cross validation for Subject 1, using $\lambda = 15$, has a cross validation error (E_{avg}) of 49.78. Table 11.10 shows the error (E_{avg}) of each subject. The average cross validation error for all 18 subjects is 69.91.

Subject	Average Model Error	Subject	Average Model Error
1	49.78	10	63.20
2	69.49	11	64.72
3	47.66	12	77.35
4	87.18	13	59.76
5	53.85	14	64.14
6	60.71	15	71.14
7	70.30	16	98.92
8	105.49	17	76.97
9	63.91	18	73.87

Table 11.10

As shown in Table 11.10, tuning the λ parameter achieves a significant improvement. Nevertheless, other approaches that decrease the number of features/predictors for the regression model may yield better results.

11.7.3 LOTO Cross Validation, 8 Channels, All Timbres

The number of predictors, for the regression function, generated by using all 14 channels is 106. The number of trials per subject is in average 86. Therefore, a contributing factor to the overfitting of the regression function is that the number of predictors surpasses the number of observations.

Reducing the number of channels decreases the number of predictors for the regression function significantly. Model selection with cross validation error as the selection criterion was used to determine which pairs of channels contain the most relevant information for the regression model. The ranking of each pair of channels was used to select the subset of eight channels that yields the best results. Figure 11.21 shows the scalp map for the eight selected channels.

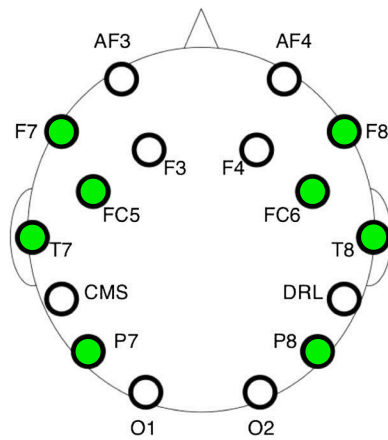


Figure 11.21: Subset of eight channels selected for the frequency experiment.

LOTO cross validation using the set of channels shown in Figure 11.21 was applied to each subject, using all four timbres. Figure 11.22 shows an example of the estimations of the training targets and the estimation of the runtime target of every iteration.

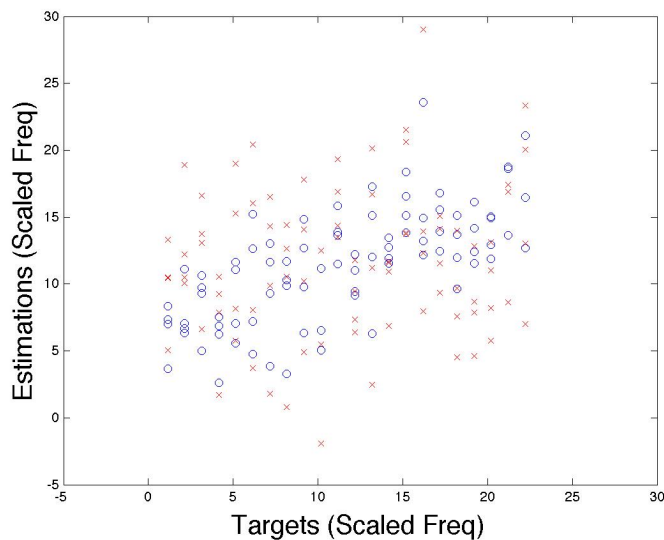


Figure 11.22: Frequency Experiment - Blue Circles show the training targets ($\mathbf{y}_{training}$) against their corresponding estimation, for the last LOTO iteration. Red crosses show the estimation of the runtime target for all LOTO iterations. The regression function of every iteration was calculated using Subject 1, $\lambda = 1$ and a frequency band of 4-7Hz.

The LOTO cross validation for Subject 1, using eight channels, has a cross validation error (E_{avg}) of 61.54. Table 11.11 shows the error (E_{avg}) of each subject. The average cross validation error for all 18 subjects is 72.20.

Subject	Cross Validation Error	Subject	Cross Validation Error
1	61.54	10	66.46
2	64.12	11	54.71
3	49.38	12	79.88
4	64.66	13	122.37
5	64.37	14	69.50
6	91.08	15	72.39
7	68.08	16	68.11
8	65.05	17	65.45
9	80.38	18	92.18

Table 11.11

As shown for the 14 channel LOTO cross validation, tuning the complexity penalty parameter of the regression improves the accuracy of the model significantly. Figure 11.23 shows the result of setting $\lambda = 15$ for the LOTO cross validation shown in Figure 11.22.

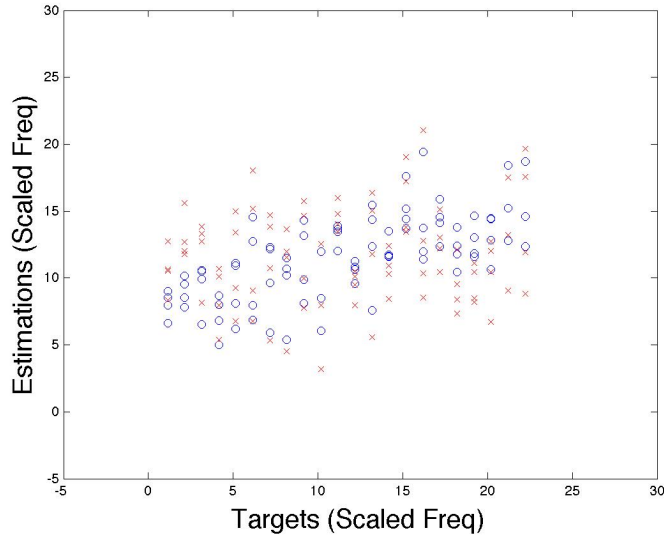


Figure 11.23: Frequency Experiment - Blue Circles show the training targets ($\mathbf{y}_{\text{training}}$) against their corresponding estimation, for the last LOTO iteration. Red crosses show the estimation of the runtime target for all the LOTO iterations. The regression function of every iteration was calculated using Subject 1, $\lambda = 15$ and a frequency band of 4-7Hz.

The LOTO cross validation for Subject 1, using eight channels and $\lambda = 15$, has a cross validation error (E_{avg}) of 47.85. Table 11.12 shows the error (E_{avg}) of each subject. The average cross validation error for all 18 subjects is 54.73.

Subject	Cross Validation Error	Subject	Cross Validation Error
1	47.85	10	51.83
2	50.75	11	48.19
3	39.12	12	63.06
4	50.04	13	64.36
5	51.47	14	53.13
6	69.30	15	56.05
7	54.83	16	51.66
8	51.37	17	50.86
9	65.62	18	65.61

Table 11.12

11.7.4 LOTO Cross Validation, 6 Channels, All Timbres

To further reduce the number of predictors for the regression model, the three pairs of channels that contained the most relevant information were selected. Figure 11.24 shows the scalp map for the six selected channels.

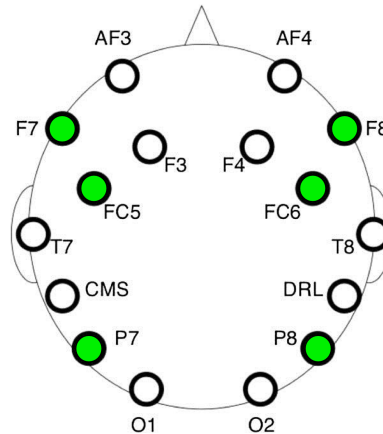


Figure 11.24: Subset of six channels selected for the frequency experiment.

LOTO cross validation using the set of channels shown in Figure 11.24 was applied to each subject, using all four timbres. Figure 11.25 shows an example of the estimations of the training targets and the estimation of the runtime target for every iteration.

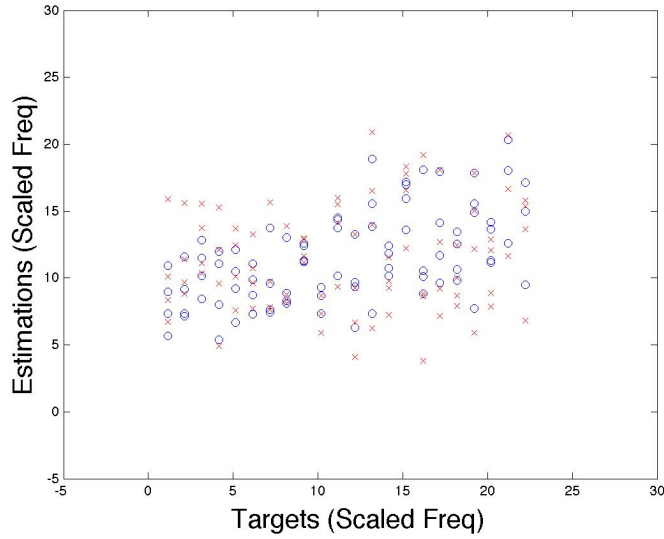


Figure 11.25: Frequency Experiment - Blue Circles show the training targets ($\mathbf{y}_{\text{training}}$) against their corresponding estimation, for the last LOTO iteration. Red crosses show the estimation of the runtime target for all LOTO iterations. The regression function of every iteration was calculated using Subject 1, $\lambda = 1$ and a frequency band of 4-7Hz.

The LOTO cross validation for Subject 1, using six channels, has a cross validation error (E_{avg}) of 47.55. Table 11.13 shows the error (E_{avg}) of each subject. The average cross validation error for all 18 subjects is 54.82.

Subject	Cross Validation Error	Subject	Cross Validation Error
1	47.55	10	51.65
2	49.62	11	48.36
3	38.98	12	57.72
4	56.52	13	58.53
5	52.90	14	53.97
6	60.25	15	58.15
7	57.35	16	71.07
8	52.90	17	55.37
9	54.13	18	61.80

Table 11.13

As shown for the 14 channels and eight channels LOTO cross validation,

tuning the complexity penalty parameter improves the accuracy of the regression model significantly. Figure 11.26 shows the results of setting $\lambda = 15$ for the LOTO cross validation shown in Figure 11.25.

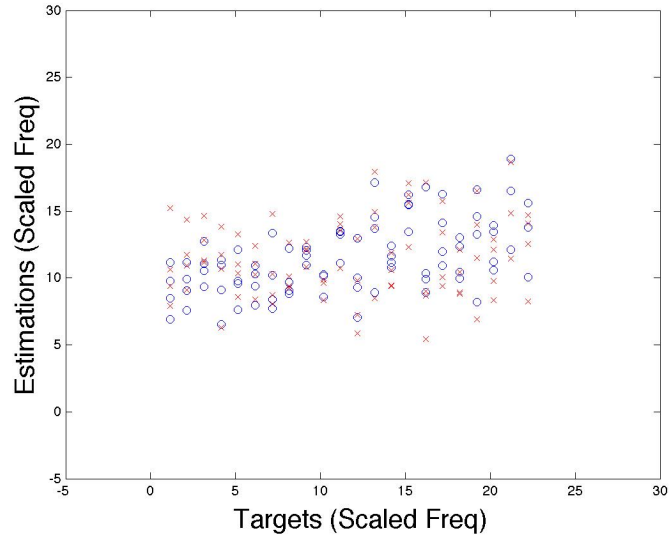


Figure 11.26: Frequency Experiment - Blue Circles show the training targets ($\mathbf{y}_{\text{training}}$) against their corresponding estimation, for the last LOTO iteration. Red crosses show the estimation of the runtime target for all the LOTO iterations. The regression function of every iteration was calculated using Subject 1, $\lambda = 15$ and a frequency band of 4-7Hz.

The LOTO cross validation for Subject 1, using six channels and $\lambda = 15$, has a cross validation error (E_{avg}) of 49.94. Table shows the error E_{avg} of each subject. The average cross validation error for all 18 subjects is 48.29.

Subject	Cross Validation Error	Subject	Cross Validation Error
1	49.94	10	47.42
2	44.86	11	43.14
3	35.22	12	51.70
4	49.39	13	48.94
5	47.99	14	48.15
6	53.39	15	52.33
7	51.72	16	52.32
8	49.95	17	48.72
9	48.79	18	55.31

Table 11.14

11.8 Separate Subjects - Summary

A summary of the results obtained for the frequency experiment using separate subjects is shown in Table 11.15.

λ	Channels' Subset Size	Average Cross Validation Error
1	14	127.12
15	14	69.91
1	8	72.20
15	8	54.73
1	6	54.82
15	6	48.29

Table 11.15

11.9 Interpolation Results

Once a linear regression was found for every subject, we can assume that interpolation can be used to find an accurate approximation of the brain's state given any loudness level or frequency. This section exemplifies the results obtained by executing Algorithm 5 for an unknown loudness level and an unknown frequency.

11.9.1 Loudness Interpolation

An interpolation for the unknown loudness level $r = 52.8$ and 14 channels was calculated using Algorithm 5. Figure 11.27 shows a visual representation of the results obtained by the interpolation.

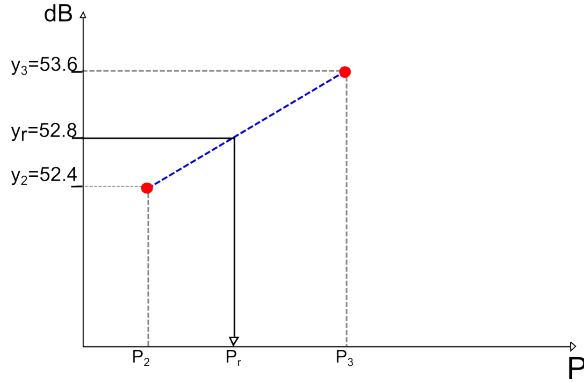


Figure 11.27: Loudness Interpolation - Unknown loudness level $r = 52.8$. Closest known loudness levels are 52.4 and 53.6 with SCM P_2 and SCM P_3 respectively.

Figure 11.28 shows the heat maps for the SCMs used in Figure 11.27.

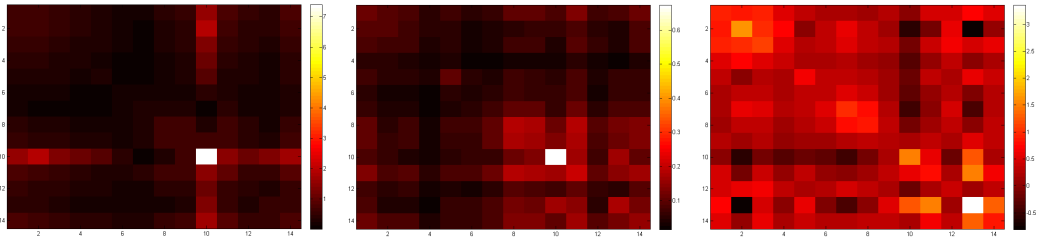


Figure 11.28: Loudness Interpolation - The leftmost figure shows the heat map of the SCM P_2 . The central figure shows the interpolated heat map of the SCM P_r . The rightmost figure shows heat map of the SCM P_3 .

11.9.2 Frequency Interpolation

An interpolation for the unknown scaled frequency $r = 2.7$ and 14 channels was calculated using Algorithm 5. Scaled frequency 2.7 corresponds to a frequency of approximately 46.77Hz, which is slightly higher than the note F#1. Figure 11.29 shows a visual representation of the results obtained by the interpolation.

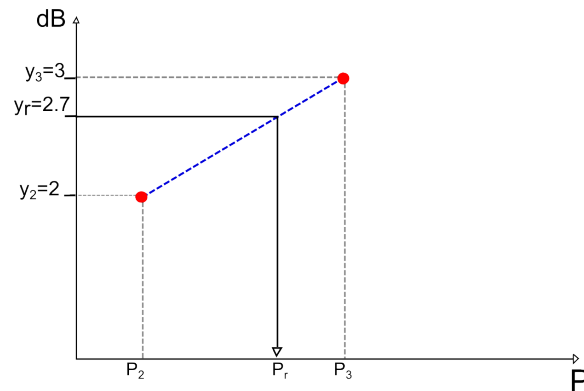


Figure 11.29: Frequency Interpolation - Unknown scaled frequency $r = 2.7$. Closest known scaled frequencies are 2 and 3 with SCM P_2 and SCM P_3 respectively.

Figure 11.30 shows the heat maps for the SCMs used in Figure 11.29.

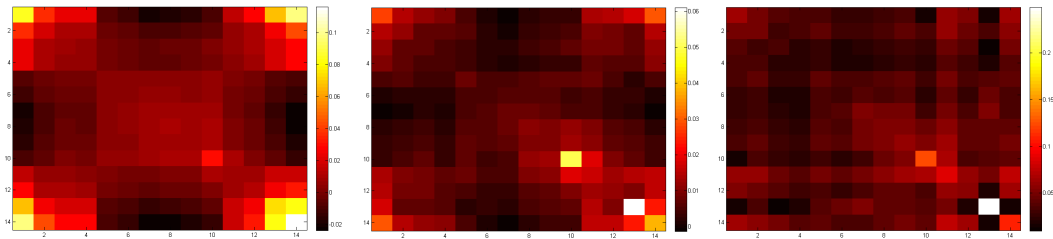


Figure 11.30: Frequency Interpolation - The leftmost figure shows the heat map of the SCM P_2 . The central figure shows the interpolated heat map of the SCM P_r . The rightmost figure shows heat map of the SCM P_3 .

Chapter 12

Conclusion

The main objective of this project is to provide a preliminary analysis of the relationship between single sound-parameter variation and the brain's response. The goal of this analysis is to find a linear regression function that models the behavior of the brain while exposed to single sound-parameter variation. If a linear regression can be found, then interpolation methods can be applied to approximate the brain's reaction to any sound parameter level within the range of known observations.

First, a general introduction to audio, music theory, and emotion research is presented. This knowledge is used for the design and implementation of two single sound-parameter variation experiments, *loudness* and *frequency*. Afterwards, the experiments are executed on 20 subjects, using the Emotiv EPOC headset.

The raw data from the experiments was pre-processed to create trials free of noise and artifacts. Successively, a measure of the distance between trials was estimated by calculating the Sample Covariance Matrices (SCM) of the trials. Since the SCMs lay in Riemannian space, Riemannian geometry and distance were used to extract the relevant features from the SCMs. The features were then used as the input for the regression model.

The first regression model was calculated for all the subjects and all the timbres using a LOSO cross validation method. Since this regression model was inaccurate, different combinations of frequency bands, timbres and channels were tested to determine if any of them contained relevant information for the regression function.

Afterwards, the dataset was separated by subject and other cross validation methods were applied. Accurate linear regression functions were found for every subject using a LOTO cross validation method. After observing

all the different regression calculations, we concluded that both loudness and frequency have a subject dependent linear relationship to the subject's brain state.

Nevertheless, the accuracy of each subject's regression model for unseen values was lower than expected due to overfitting of the regression function. The original assumption of the project was that a linear regression between single sound-parameter variation and the brain's reaction could be found for all subjects. Therefore, the number of trials per subject is limited. This causes the number of predictors of the regression function to be higher than the number of observations per subject, thus, overfitting.

Tuning of the complexity penalty parameter λ of the regression function, decreases the number of predictors and yields significant improvements for the estimation of unseen values. Model selection using cross validation error as a selection criterion was executed for selecting the best channels and the best frequency band. A reduction in the number of channels used for the SCMs decreases the number of predictors drastically and also improves the accuracy of the regression functions.

For the loudness experiment, the smallest average cross validation error for all subjects is 221.59. This LOTO cross validation method uses a subset of 6 channels, $\lambda = 15$ and a frequency band of 4-7Hz. For the frequency experiment, the smallest average cross validation error for all subjects is 48.29. This LOTO cross validation method uses a subset of 6 channels, $\lambda = 15$ and a frequency band of 4-7Hz.

Since the cross validation error represents an average MSR, the square root of the average cross validation error $\sqrt{E_{avg}}$ represents a measure of distance. This measure can be used to compare the results obtained for the loudness and frequency experiments.

The square root of the minimum average error for the loudness experiment represents a distance of 14.88dB. If we compare this distance to the scale used for the loudness experiments (45-110dB) then the error represented by 14.88dB is approximately 22%, $(14.88/(110 - 45)) \approx 0.22$. The square root of the minimum average error for the frequency experiment represents a distance of 6.94 scaled frequencies. If we compare this distance to the scale used for the frequency experiments (1-22 scaled frequencies) then the error represented by 6.94 scaled frequencies is approximately 33%, $(6.94/(22 - 1)) \approx 0.33$.

As stated before, the high percentage of error for unseen values is due to overfitting. Therefore, an experiment designed to obtain a higher num-

ber of trials per subject might yield a more accurate and general regression function. Nevertheless, the fact that a linear regression could be found for every subject enables the assumption that interpolation can be used to estimate the SCM of any audio parameter level within the known observations.

Chapter 13

Future Works

The methodologies and theories used throughout this project open multiple possibilities for continuing the analysis of the effect that sound has in the human brain.

The first element needed to improve the models proposed in this project, is an experiment that generates a greater number of trials. Since both the loudness and frequency parameters were found to be subject dependent, a more extensive dataset per subject may yield significantly better estimations for unseen values.

Other sound parameters could be tested to determine if they have a linear behavior. For example, changes in the ADSR envelope of a sound could be analyzed to further understand how the brain processes surprise and unexpected changes in sounds.

More extensive studies could analyze the emotional implications of the model selection executed for this project. For example, most of the relevant information for the regression model is found in the frontal cortex, but a linear model could not be found for frequency without the pair of channels P7 and P8 located in the Parietal Lobe. This could be studied more in-depth to determine if certain emotional states are induced by the sound parameters.

A systematic analysis of all different sound parameters might yield enough information to be used by the entertainment industry. By analyzing the effect each parameter and combination of parameters have in the brain's state, a generalization of the emotional effects of sounds can be achieved and used to induce emotion in an audience.

In the gaming industry, using BCI technology to achieve an understanding and control of the emotional state of the user can be used to create a

completely different, more immersive, gaming experience. In the music therapy field, a deeper understanding of the brain's emotional reaction to sound could be used to improve the mood of the patient or induce a beneficial mental state.

Furthermore, the information on how the brain perceives a certain level of loudness or frequency can help obtain a more objective understanding of how the human brain processes sound and music.

Bibliography

- [1] W. J. Freeman, *Mass action in the nervous system: examination of the neurophysiological basis of adaptive behavior through the EEG*. MIT Press, 2006.
- [2] B. Blankertz, R. Tomioka, S. Lemm, M. Kawanabe, and K.-R. Muller, “Optimizing spatial filters for robust EEG single-trial analysis,” *Signal Processing Magazine, IEEE*, vol. 25, no. 1, pp. 41–56, 2008.
- [3] A. R. Gregersen, A. Trevisi, A. R. Rivera, D. A. H. Jensen, J. F. Faarkrog, and N. Andersen, “An Examination of Signal Processing and Classification Methods for Brain Computer Interfaces,” semester project, Aalborg University, 2013.
- [4] B. Graimann, B. Allison, and G. Pfurtscheller, “Brain–Computer Interfaces: A Gentle Introduction,” *Brain–Computer Interfaces, Revolutionizing Human–Computer Interaction*, no. 1, pp. 1–27, 2010.
- [5] A. Barachant, S. Bonnet, M. Congedo, and C. Jutten, “Multiclass Brain–Computer Interface Classification by Riemannian Geometry,” *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 4, pp. 920–928, 2012.
- [6] Y.-H. Yang, Y.-C. Lin, Y.-F. Su, and H. H. Chen, “A Regression Approach to Music Emotion Recognition,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 16, pp. 448–457, 2008.
- [7] J. A. Russell, “Affective Space is Bipolar,” *Journal of Personality and Social Psychology*, vol. 37, pp. 345–356, 1979.
- [8] R. E. Thayer, *The Biopsychology of Mood and Arousal*. Oxford University Press, 1989.
- [9] T. Garner, “Game Sound from Behind the Sofa: An Exploration into the Fear Potential of Sound and Psychophysiological Approaches to Audio-centric, Adaptive Gameplay,” phd thesis, University of Aalborg, Aalborg University, 2012.

- [10] I. Ekman and R. Kajastila, “Localisation Cues Affect Emotional Judgements – Results from a User Study on Scary Sound. Proc.,” *AES 35th Conference on Audio for Games*, feb 2009.
- [11] G. Loy, *Musimathics: The mathematical foundations of music*. Academic Press, 1975.
- [12] H. Stephen, *An Introduction to the Perception of Auditory Events*. A Bradford Book, 1989.
- [13] D. Howard and J. Angus, *Acoustics and Psychoacoustics*. Audio/Music technology, Focal Press, 2006.
- [14] R. Parncutt, *Harmony: A Psychoacoustical Approach*. Springer-Verlag, 1989.
- [15] J. Cho, E. Yi, and G. Cho, “Physiological responses evoked by fabric sounds and related mechanical and acoustical properties,” *Textile Research Journal*, vol. 71, no. 12, pp. 1068–1073, 2001.
- [16] D. Västfjäll, “Emotional Reactions to Sounds without Meaning,” *Psychology*, vol. 2, no. 8, pp. 606–609, 2012.
- [17] D. Bach, J. Neuhoff, W. Perrig, and E. Seifritz, “Looming sounds as warning signals: The function of motion cues,” *International Journal of Psychophysiology*, vol. 74, no. 1, pp. 28–33, 2009.
- [18] E. Asutay and D. Västfjäll, “Perception of Loudness Is Influenced by Emotion,” *PLoS ONE*, vol. 7, no. 6, 2011.
- [19] E. H. Seigel and J. K. Stefanucci, “A little bit louder now: Negative affect increases perceived loudness,” *Emotion*, vol. 11, no. 4, 2011.
- [20] D. R. Langers, P. van Dijk, E. S. Schoenmaker, and W. H. Backes, “fMRI activation in relation to sound intensity and loudness,” *NeuroImage*, vol. 35, pp. 709–718, 2007.
- [21] F. Hagman, *Emotional response to sound*. Master thesis, Chalmers University of Technology, Chalmers University of Technology, 2010.
- [22] J. Edworthy, S. Loxley, and I. Dennis, “Improving Auditory Warning Design: Relationship between Warning Sound Parameters and Perceived Urgency,” *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 33, no. 2, pp. 205–231, 1991.
- [23] J. Parker and J. Heerema, “Audio Interaction in Computer Mediated Games,” *International Journal of Computer Games Technology*, vol. 2008, p. Article ID178923, 2007.

- [24] S. Moncrieff, “Affect computing in film through sound energy dynamics,” in *In Proc. of ACM Multimedia’ 01*, 2011.
- [25] M. Grimshaw, “The Audio Uncanny Valley: Sound, Fear and the Horror Game,” *Audio Mostly 4th Conference on Interaction with Sound, Glasgow, 2nd–3rd September*, 2009.
- [26] M. Grimshaw, “The resonating spaces of first-person shooter games,” *The 5th International Conference on Game Design and Technology, Game Design Technology*, 2007.
- [27] A. Jiménez, P. Larsson, A. Våljamäe, D. Västfjäl, and M. Kleiner, “When Room Size Matters: Acoustic Influences on Emotional Responses to Sounds,” *Emotion*, vol. 10(3), pp. 416–422, 2010.
- [28] M. Breinbjerg, “The Aesthetic Experience of Sound – staging of Auditory Spaces in 3D computer games,” *Aesthetics of Play conference in Bergen, Norway*, 14-15 oct. 2005.
- [29] L. Barrington, M. J. Lyons, D. Diegmann, and S. Abe, “Ambient Display using Musical Effects,” *IUI ’06 Proceedings of the 11th international conference on Intelligent user interfaces*, pp. 372–374, 2006.
- [30] I. Ekman, “Psychologically Motivated Techniques for Emotional Sound in Computer Games,” *Audio Mostly Pitea, Sweden*, oct 2008.
- [31] K. Hevner, “The affective character of the major and minor modes in music,” *The American Journal of Psychology*, vol. 47, pp. 103–108, 1935.
- [32] K. Hevner, “The affective value of pitch and tempo in music,” *The American Journal of Psychology*, vol. 49, pp. 621–630, 1937.
- [33] K. N. Goydke, E. Altenmüller, J. Möller, and T. F. Münte, “Changes in emotional tone and instrumental timbre are reflected by the mismatch negativity,” *Brain Res Cogn Brain Res*, vol. 21, pp. 351–9, month 2004.
- [34] P. N. Juslin, “Communicating emotion in music performance: a review and theoretical framework,” *Music and Emotion—Theory and Research*, pp. 309–337, 2001.
- [35] C. Laurier, O. Lartillot, T. Eerola, and P. Toiviainen, “Exploring Relationships between Audio Features and Emotion in Music,” *ESCOM, Conference of European Society for the Cognitive Sciences of Music*, aug 2009.
- [36] E. Rapoport, “Singing, mind and brain—unit pulse, rhythm, emotion and expression,” *Music, Gestalt, and Computing: Studies in Cognitive and Systematic Musicology*, pp. 451–468, 1997.

- [37] C. P. Niemic, “A Theoretical and Emperical Review of Psychophysiological Studies of Emotion,” *Journal of Undergraduate Research*, vol. 1, no. 1, pp. 15–18, 2002.
- [38] K. Trochidis and E. Bigand, “EEG-based emotion perception during music listening,” *International Conference on Music Perception and Cognition (ICMPC), Thessaloniki*, 2012.
- [39] R. J. Davidson, “EEG measures of cerebral asymmetry: Conceptual and Methodological issues,” *International Journal of Neuroscience*, vol. 39, pp. 71–78, 1988.
- [40] R. J. Davidson, P. Eckman, C. Saron, J. Senulis, and W. W. Friegenn, “Approach-Withdrawal and Cerebral Asymmetry: Emotional expression and Brain Physiology,” *Journal of Personality and Social Psychology*, vol. 158, no. 2, pp. 330–341, 1990.
- [41] L. Schmidt and L. Trainor, “Frontal brain electrical activity (EEG) distinguishes valence and intensity of musical emotion,” *Cognition and Emotion*, vol. 15, no. 4, pp. 487–500, 2001.
- [42] D. Sammler, M. Grigutsch, T. Fritz, and S. Koelsh, “Music and Emotion: Electrophysiological correlates of the processing of pleasant and unpleasant music,” *Psychophysiology*, vol. 44, pp. 293–304, 2007.
- [43] R. Oostenveld and P. Praamstra, “The five percent electrode system for high-resolution eeg and erp measurements,” *Clinical Neurophysiology*, vol. 112, no. 4, pp. 713–719, 2001.
- [44] TransCranialTechnologies, *10/20 System Positioning - Manual*. http://www.trans-cranial.com/local/manuals/10_20_pos_man_v1_0_pdf.pdf, 2012.
- [45] M. Duvinage, T. Castermans, M. Petieau, T. Hoellinger, and T. Dutoit, *RESEARCH Open Access Performance of the Emotiv Epoc headset for P300-based applications*.
- [46] P. L. Nunez, B. M. Wingeier, and R. B. Silberstein, “Spatial-temporal structures of human alpha rhythms: Theory, microcurrent sources, multiscale measurements, and global binding of local networks,” *Human brain mapping*, vol. 13, no. 3, pp. 125–164, 2001.
- [47] T. Harmony, T. Fernández, J. Silva, J. Bernal, L. Díaz-Comas, A. Reyes, E. Marosi, M. Rodríguez, and M. Rodríguez, “EEG delta activity: an indicator of attention to internal processing during performance of mental tasks,” *International Journal of Psychophysiology*, vol. 24, no. 1, pp. 161–171, 1996.

- [48] W. Klimesch, "EEG alpha and theta oscillations reflect cognitive and memory performance: a review and analysis," *Brain research reviews*, vol. 29, no. 2, pp. 169–195, 1999.
- [49] J. L. Cantero, M. Atienza, R. Stickgold, M. J. Kahana, J. R. Madsen, and B. Kocsis, "Sleep-dependent θ oscillations in the human hippocampus and neocortex," *The Journal of Neuroscience*, vol. 23, no. 34, pp. 10897–10903, 2003.
- [50] S. Palva and J. M. Palva, "New vistas for α -frequency band oscillations," *Trends in Neurosciences*, vol. 30, no. 4, pp. 150 – 158, 2007.
- [51] J. A. Pineda, "The functional significance of mu rhythms: translating "seeing" and "hearing" into "doing"," *Brain Research Reviews*, vol. 50, no. 1, pp. 57–68, 2005.
- [52] Y. Zhang, Y. Chen, S. L. Bressler, and M. Ding, "Response preparation and inhibition: the role of the cortical sensorimotor beta rhythm," *Neuroscience*, vol. 156, no. 1, pp. 238–246, 2008.
- [53] H. E. Wolfe, *Introduction to Non-Euclidean Geometry*. Mill Press, 2007.
- [54] S. Christina, *What is Riemannian Geometry ?* <http://comet.lehman.cuny.edu/sormani/research/riemgeom.html>, 2002.
- [55] S. Mario, *Geometrie Non Euclidee*. http://www.mariosandri.it/scuola/ssis/Geometrie_noneuclidee.pdf, 2005.
- [56] G. Martini, "Geodetiche su varietà Riemanniane," master thesis, Università di Bologna, Università di Bologna, 2010.
- [57] E. W. Weisstein, *Frobenius Norm- From MathWorld- A Wolfram Web Resource*. <http://mathworld.wolfram.com/FrobeniusNorm.html>.
- [58] G. Golub and C. Van Loan, *Matrix Computation, 3rd ed.* Baltimore MD., 1996.
- [59] A. Barachant, S. Bonnet, M. Congedo, and C. Jutten, "Classification of covariance matrices using a Riemannian-based kernel for BCI application," *Neurocomputing*, vol. 112, pp. 172–178, 2013.
- [60] E. W. Weisstein, *"L2-Norm." From MathWorld-A Wolfram Web Resource*. <http://mathworld.wolfram.com/L2-Norm.html>.
- [61] A. Barachant, S. Bonnet, M. Congedo, and C. Jutten, "Riemannian geometry Applied to BCI classification," *Latent Variable Analysis and Signal Separation*, pp. 629–636, 2010.

- [62] W. Forstner and B. Moonen, "A Metric for Covariance Matrices," *Geodesy-The Challenge of the 3rd Millennium*, pp. 299–309, 2003.
- [63] M. Moakher, "A Differential Geometric Approach To The Geometric Mean of Symmetric Positive-Definite Matrices," *SIAM J. Matrix Anal. Appl.*, vol. 26, no. 3, pp. 735–747, 2005.
- [64] O. Tuzel, F. Porikli, and P. Meer, "Pedestrian Detection Via Classification on Riemannian Manifolds," *Pattern Analysis and Machine Intelligence*, vol. 30, no. 10, pp. 1713–1727, 2008.
- [65] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer Science, 2006.
- [66] S. H. Brown, *Multiple Linear Regression Analysis: A Matrix Approach with MATLAB*. <http://ajmonline.org/2009/brown.pdf>, 2009.
- [67] E. W. Weisstein, "L1-Norm." *From MathWorld—A Wolfram Web Resource*. <http://mathworld.wolfram.com/L1-Norm.html>.
- [68] A. E. Hoerl and R. W. Kennard, "Ridge Regression: Biased Estimation for Nonorthogonal Problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.
- [69] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. Cambridge University Press., 2004.
- [70] A. Goh, "Unsupervised riemannian clustering of probability density functions," *Machine Learning and Knowledge Discovery in Databases*, vol. 5211, pp. 377–392, 2008.
- [71] J. Bowers, *Bowers-Wilkins DM7 MKII*. http://www.bowers-wilkins.com/Downloads/Product/Archive/ENG_FT00063_DM7-MK2_info_sheet.pdf.
- [72] A. Delorme, T. Fernsler, H. Serby, and S. Makei, *EEGLAB Tutorial*. <ftp://sccn.ucsd.edu/pub/eeglabtutorial15.00b.pdf>, 2006.
- [73] R. Daniels, *Approximation Methods for Electronic Filter Design*. New York: McGraw-Hill, 1974.
- [74] MATLAB, *version 7.12.0 (R2011a)*. Natick, Massachusetts: The MathWorks Inc., 2011.
- [75] A. Barachant, *Covariance Toolbox*. <https://github.com/alexandrebarachant/covariancetoolbox>, 2013.