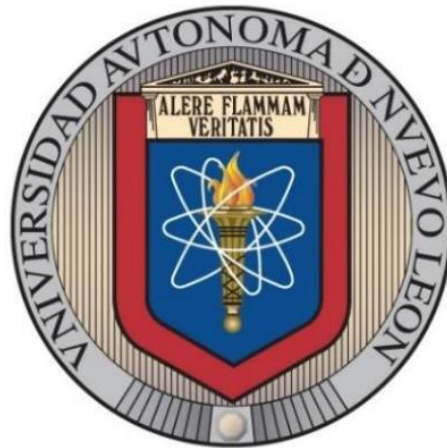


UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA



**ESTIMATION OF LOWER-LIMB JOINT TORQUES FROM EEG SIGNALS
OF GAIT INITIATION MOVEMENTS USING A MACHINE LEARNING
SCHEME**

BY

LUCERO ESMERALDA ALVARADO RUIZ

**IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF MASTER OF SCIENCE IN
ELECTRICAL ENGINEERING**

SEPTEMBER, 2019

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA
SUBDIRECCIÓN DE ESTUDIOS DE POSGRADO



**ESTIMATION OF LOWER-LIMB JOINT TORQUES FROM EEG SIGNALS
OF GAIT INITIATION MOVEMENTS USING A MACHINE LEARNING
SCHEME**

BY

LUCERO ESMERALDA ALVARADO RUIZ

**IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF MASTER OF SCIENCE IN
ELECTRICAL ENGINEERING**

SEPTEMBER, 2019

Universidad Autónoma de Nuevo León
Facultad de Ingeniería Mecánica y Eléctrica
Subdirección de Estudios de Posgrado

Los miembros del Comité de Tesis recomendamos que la Tesis “Estimation of lower-limb joint torques from EEG signals of gait initiation movements using a machine learning scheme”, realizada por la alumna Lucero Esmeralda Alvarado Ruiz, con número de matrícula 1543412, sea aceptada para su defensa como requisito parcial para obtener el grado de Maestra en Ciencias de la Ingeniería Eléctrica.

El Comité de Tesis

Dra. Griselda Quiroz Compeán
Asesora

Dr. Luis Martín Torres Treviño
Revisor

Dr. Romeo Sánchez Nigenda
Revisor

Vo. Bo.

Dr. Simón Martínez Martínez

Subdirección de Estudios de Posgrado



*A mis padres
que abrazan con amor
cada uno de mis sueños.*

CONTENTS

Acknowledgments	IV
Contents	VI
List of Figures	VIII
List of Tables	XII
List of Algorithms	XIII
Abstract	XIV
1 Introduction	1
1.1 Motivation	2
1.2 Literature Survey	2
1.3 Problem Statement	4
1.4 Hypothesis	4
1.5 Aim and Objectives	5
1.6 Outline of the Thesis	5
2 Theoretical Framework	6
2.1 Physiological Context	6
2.1.1 Movements of the lower-Limb During Walking	6
2.1.1.1 Hip Joint	7
2.1.1.2 Knee Joint	8
2.1.1.3 Ankle Joint	8
2.1.2 Neural Supply of the lower-Limb	9
2.1.2.1 EEG	10
2.2 Machine Learning (ML) Context	12
2.2.1 Time Series Matching	12
2.2.1.1 Dynamic Time Warping (DTW)	12
2.2.2 Clustering	15
2.2.2.1 Balanced Iterative Reducing and Clustering (BIRCH)	15
2.2.2.2 Consistency of Cluster Assignment	17

2.2.2.3	Density of Clusters	19
2.2.3	Regression	19
2.2.3.1	K-Nearest Neighbors (KNN)	19
2.2.3.2	Multi-layer Perceptron (MLP)	20
2.2.3.3	Measures of Goodness of Fit	22
2.2.4	Optimization	24
2.2.4.1	Genetic Algorithm (GA).....	24
3	Proposed Scheme	26
3.1	Dataset and Pre-processing	26
3.1.1	Gait Initiation Movements	27
3.1.2	Torque Acquisition	28
3.1.3	EEG Acquisition	29
3.2	Optimization Stage	31
3.2.1	Clusterized KNN (C-KNN) for Prototyping	32
3.2.2	Channel Selection	36
3.3	Torque Estimation	37
4	Results	38
4.1	Implementation details	38
4.2	Prototypes Extracted	39
4.2.1	Consistency of Extraction	39
4.2.2	Comparison to Original Torques	41
4.3	Selected Channels	43
4.4	Estimated Torques	46
5	Discussion	51
5.1	On the Extraction of Prototypes	51
5.2	On the Selection of Channels	52
5.3	On the Estimation of Torque	52
6	Conclusions	54
	Bibliography	55

LIST OF FIGURES

2.1	Phases and divisions of the gait cycle according to [12]. Adapted from [11]	7
2.2	Anatomical position and vertical planes of the human body.	7
2.3	Movements of the hip joint.	8
2.4	Movements of the knee joint.	8
2.5	Movements of the ankle joint.	9
2.6	A single plane projection of the head with electrode placement according to the international 10-20 system.	10
2.7	Effect of the bound constraint in the generation of the warping path in the DTW algorithm.	13
2.8	Effect of the arbitrary assignation of number of clusters. (a) When the number of clusters is 3, the optimal clustering is reached. (b) and (c) shows a deficient clustering, consequence of an arbitrary assignment of the number of clusters.	15
2.9	Structure of a CF-tree.	16
2.10	Example of a silhouette analysis on sample data with three clusters. (a) Shows the silhouette plot of the cluster assignment, two of the values of cluster 0 (<i>blue dots and blue bars</i>) might be wrongly assigned. (b) A visualization of the clustered data.	18
2.11	Architecture of a MLP.	20
3.1	General diagram of the proposed ML scheme for torque estimation. . .	26

4.14 R^2 and SNR statistics of the top 5 estimations, according to R^2 , of prototype torques with respect to joint (<i>pairs of axes</i>), type of movement (<i>blue, green or red dots</i>) and left or right limb (<i>light and dark dots</i>).	49
4.15 R^2 and SNR statistics of the top 5 estimations, according to R^2 , of prototype torques with respect to joints (<i>pairs of axes</i>) and subjects (<i>dot color</i>).	50

LIST OF TABLES

4.1	Summary statistics of the percentage of extracted prototypes and number of selected channels.	44
-----	--	----

LIST OF ALGORITHMS

2.1	Pseudocode of DTW	14
2.2	Pseudocode of BIRCH	17
2.3	Pseudocode of KNN search.	20
2.4	Pseudocode of the MLP with Adam optimization	23
2.5	Main phases of the GA	25
3.1	Pseudocode of C-KNN.	35

ABSTRACT

Lucero Esmeralda Alvarado Ruiz.

Candidate for the Degree of Master of Science in Electrical Engineering.

Universidad Autónoma de Nuevo León.

Facultad de Ingeniería Mecánica y Eléctrica.

Title of the study:

Estimation of lower-limb joint torques from EEG signals
of gait initiation movements using a machine learning scheme.

Number of pages: 58.

RESUMEN La capacidad de realizar tareas cotidianas juega un papel fundamental en la forma en que las personas se perciben a sí mismas como miembros aportadores a la sociedad. Esta percepción puede cambiar drásticamente cuando se producen pérdidas de funciones en el cuerpo, sobre todo si éstas implican la pérdida de una extremidad, lo que es particularmente preocupante si tomamos en cuenta que la pérdida de movilidad en las extremidades inferiores es una de las discapacidades más comunes. Las interfaces cerebro computadora, Brain-Computer Interfaces (BCIs), surgen como un intento de devolver a las personas con discapacidad la movilidad que alguna vez tuvieron. Uno de sus objetivos es la estimación de manera precisa de variables continuas de la locomoción humana, tales como fuerzas musculares, posiciones articulares, posiciones cartesianas y pares articulares, a partir de mediciones neuronales no invasivas, como las realizadas por

medio de electroencefalografía (EEG). Al diseñar esquemas que integren la adquisición de señales EEG y algoritmos para su procesamiento, se puede lograr el objetivo antes mencionado. Sin embargo, todavía no existe un método factible para adquirir y procesar estas señales de una manera que permita estimaciones confiables; además, no hay propuestas para la estimación de variable cinética de extremidades inferiores, como pares articulares, durante la ejecución de movimientos fuera del ciclo de la marcha, tales como movimientos de inicio de la marcha. Por lo tanto, esta tesis presenta un esquema para estimar pares articulares de las extremidades inferiores a partir de señales EEG adquiridas durante la ejecución de movimientos de inicio de la marcha. Este esquema se aplica para optimizar un subconjunto de canales de EEG a partir del cual se lleva a cabo una extracción de la información más relevante del conjunto de datos. Estos datos extraídos, llamados prototipos y obtenidos mediante un algoritmo propuesto, se utilizan para entrenar y validar el modelo de regresión que se encarga de estimar los pares articulares. Los resultados muestran estimaciones exitosas y que los canales de EEG más frecuentemente seleccionados son consistentes con las regiones del cerebro que se sabe se activan durante tareas motrices. Los datos extraídos se pueden utilizar para un posterior análisis que tenga el fin de caracterizar procesos neuronales.

ABSTRACT The ability to perform physical tasks of daily living plays a critical role in the way people perceives themselves as contributors to society. This perception can be drastically modified when loss of function in the body occurs, specially if this involves the loss of a limb. This is particularly worrying as loss of mobility in the lower limbs is one of the commonest physical impairments. Brain-Computer Interfaces (BCIs) surge as an attempt to return the mobility impaired people once had. One of the aims BCIs have is the accurate estimation of continuous variables of the human locomotion, such as muscular forces, articular positions, cartesian positions and joint torques, from non-invasive neural recordings, such as Electroencephalographic (EEG) signals. By designing schemes that integrate acquisition of EEG signals and processing algorithms, the aforementioned aim may be achievable. However, there is still no feasible method

to acquire and process these signals in a way that allows reliable estimations; moreover, there is lack of proposes for the estimation of kinetic variables of the lower limb, such as joint torques, during the execution of movements outside the gait cycle, such as gait initiation movements. Hence, this thesis presents a scheme for estimating lower-limb joint torques from EEG acquired during the execution of gait initiation movements. The scheme is applied to optimize the subset of EEG channels from which the extraction of the most relevant information of the dataset is carried out. This extracted data, referred as prototypes and obtained with a proposed algorithm, is used to train and test the regression model that estimates the desired joint torques. Results show that successful estimations are obtained, and that the most selected EEG channels are consistent with the regions of brain known to be activated during motor tasks. Extracted data can be used for further analysis in order to characterize neural processes.

Advisor's signature: _____
PhD. Griselda Quiroz Compeán

CHAPTER 1

INTRODUCTION

As part of the everyday life, people must perform certain tasks or activities that involve interaction with the environment, which mainly include walking and touching things. However, the execution of such activities becomes especially difficult for people with mobility impairments. Several schemes for rehabilitation and restoration of mobility have been developed over the last years to address this problem. Conventional rehabilitation therapies consist of performing exercises during certain time and a certain number of sessions, depending on the severity of the mobility loss. Although there is a wide spectrum of therapies, there is no guarantee that patients will be able to regain the mobility they once had, and physical therapies are not useful in cases when total loss of communication between brain and limbs has occurred.

For those cases, BCIs have been developed during the last years as an alternative way of communication between brain and environment through the use of assistive devices, such as prostheses, wheelchairs and exoskeletons. BCIs provide the user with some sort of control over the assistive device, which can be used to classify or to estimate movement intentions by using neural activity.

The focus of this thesis is the estimation of kinetic variable in the lower-limb from non-invasive recordings of brain signals, acquired during the execution of gait initiation movements.

1.1 MOTIVATION

The focus on lower-limbs is motivated by two reasons. First, the impairment of the lower-limbs is the most common type of disability in USA and Mexico [1, 2]. Second, most of the schemes for rehabilitation and restoration of mobility are focused on the upper-limb, and those who analyze the lower-limb are focused on the gait cycle. Therefore, analyzing the initiation of movement offers an opportunity to contribute to the understanding of locomotion and the use of BCIs to restore it.

Studying the process of initiation of the gait cycle provides insights about how representative the information contained in the measurable signals of the brain is for the human locomotion, which is the basis behind the daily tasks one performs as part of the interaction with the environment. Such knowledge enhances the development of motor assistance technologies, since it will allow prosthetic limbs to be controlled with neural information, acquired preferably from non invasive methods. It also represents a tool for a better diagnosis of motor disabilities, identification of its causes and subsequent treatment.

Regarding the approach taken, machine learning has shown to be reliable for solving real life problems in which the underlying processes are not extensively known or measurable, by searching for hidden structures and information in an available dataset.

1.2 LITERATURE SURVEY

The acquisition of neural activity for the control of BCIs has gained great interest since Fitzsimmons et al. [3] used neural signals for the estimation of angles of hip, knee and ankle joints of macaques during the cycle of bipedal walking. Their decoding method used linear Wiener filters for the estimation of kinematics. However, the neural signals were acquired via the invasive method Electroencephalography (EEG), which requires a surgeon for the placement of intracortical electrodes. As the ethical implications of such procedures becomes of special concern for its application in the human body, later attempts of estimating variable of movement from brain signals were carried out using recordings of

Electroencephalographic (EEG) signals, which consist in the placement of electrodes in the scalp. Thus this non-invasive method became one of the most used methods for acquisition of human brain signals with the aim of decoding movement.

Starting from this, a later result of Presacco et al. [4] showed estimation of joint angles of the hip, knee and ankle joints from EEG signals by using linear Wiener filters with estimations above the chance level. Although this work set the tone for subsequent research of kinematic variable estimation from non-invasive methods, it should be noted that the limitations of this study are the use of a very restricted range of EEG frequencies (0.1 to 2 Hz) and the focus in a defined gait cycle, so that the estimated signal has sinusoidal-like waveform. The use of the coefficient of correlation for assessing the goodness of fit for this type of waveform could lead to a misinterpretation of the results [5].

In contrast to the estimation of continuous kinematic variable, another kind of decodification of movement from EEG signals is the classification of tasks. Dong Liu et al. [6] developed a BCI which attempts to detect the intention of ankle movement from low frequencies of EEG by using a support vector machine. Kilicarslan et al. [7] decoded intentions of right-turning and sit-rest-stand motions of paraplegic subjects with high offline accuracy from low frequencies of EEG recordings by using Gaussian classifiers.

Studies that went far from the low frequencies of EEG where those of Gwin and Ferris [8], who distinguished ankle tasks from knee tasks with high accuracy from recordings of α and β modulations of EEG by using independent component analysis and Bayesian classifiers. Similarly, Seeber et al. [9] reconstructed gait cycle patterns from low gamma modulations (24-40 Hz) of EEG by using Morlet wavelets. The notable difference of the aforementioned studies compared to the vast majority of other studies, is the usage of a wider range of frequencies of the EEG signals, based on established results of neurosciences [10]. Such results states that oscillatory cortical activity in of 8 to 30 Hz present desynchronization during the execution of movement.

The most common methods for extraction of information from the EEG signals

in the previous studies are those of spectral analysis. Another kind of information discrimination is done by selecting a subset of electrodes in the EEG signals, either arbitrarily or by discriminating zones of interest in the scalp. The drawbacks of these methods is the lacking of a deeper analysis of the significance of each electrode for the overall dataset. Therefore, this thesis focuses on the extraction of subsets of EEG prior to the decoding stage. The selection of electrodes is carried out in such a way that attempts are made to extract the most relevant information from the process of generation of movement in the lower-limb.

1.3 PROBLEM STATEMENT

As seen in the previous section, there are schemes for the reconstruction of movements of the lower-limb from EEG signals. These schemes decode kinematic variables based on the lower frequencies of neural activity during the gait cycle using linear methods. However, it is a matter of importance to consider the frequency range recognized as the fluctuating one during motor activity, furthermore to consider movements of the initiation of the gait cycle and a possible non-linearity of the relationship between relevant variables, thus exploiting the available information of the process. In addition, the estimation of kinetic variable is of special interest as it is related with both the muscular force and the kinematics of the lower-limb.

1.4 HYPOTHESIS

Based on insights given by previous studies in the field of BCIs, and by using ML techniques for identifying data prototypes and relating datasets, it is possible to find a relationship between EEG recordings and the joint torques generated in the lower-limb during the execution of gait initiation movements. Such relationship is expected to allow the continuous reconstruction of joint torques from only EEG signals.

1.5 AIM AND OBJECTIVES

The aim of this study is to propose a scheme for relating the neuronal and mechanical information of the lower limb acquired during the execution of gait initiation movements through an analysis of the data based on ML techniques. The specific objectives of this thesis are:

- To propose a ML scheme for identifying prototypes of the available data that represent the overall process.
- To implement optimization algorithms for finding optimal subsets of EEG channels that allow a better prototyping of the data.
- To implement regression algorithms for finding the relationship between identified prototypes of EEG recordings and joint torques of the lower-limb during gait initiation movements.

1.6 OUTLINE OF THE THESIS

The rest of this document is organized as follows. In Chapter 2, the reader is provided with a brief theoretical framework on physiological definitions and ML methods that will be mentioned later. Then, in Chapter 3, the proposed scheme to solve the estimation of torque from EEG signals using the methods described in Chapter 2 is explained. The results obtained for each defined objective through the application of the proposed scheme are presented in Chapter 4; these results are discussed in depth in Chapter 5. Finally, overall conclusions on the contributions of this work and recommendations for future ones are discussed in Chapter 6.

CHAPTER 2

THEORETICAL FRAMEWORK

In this chapter, the definitions, algorithms, methods and measures used in this thesis are briefly described. The theoretical framework is divided into two contexts: the physiological and the computational one.

2.1 PHYSIOLOGICAL CONTEXT

In this section, definitions for a basic understanding of the human locomotion during walking and the processes that allow it to occur are described.

2.1.1 MOVEMENTS OF THE LOWER-LIMB DURING WALKING

In order to describe the movements of joints in the lower-limb that are executed during walking, it is necessary to introduce the concept of gait cycle.

GAIT CYCLE. The gait cycle is the repetition of a sequence of movements that are executed during walking. The phases of the gait cycle, according to the floor contact by the two feet, are illustrated in Figure 2.1 (adapted from [11]). Then, several movements are observed in the hip, knee and ankle joints during the gait cycle and variations of it, such as stair climbing, backward walking and lateral walking. The following descriptions start from the assumption of the anatomical position as the initial position of joints. The anatomical position, as well as the relevant anatomical planes, can be seen in Figure 2.2.

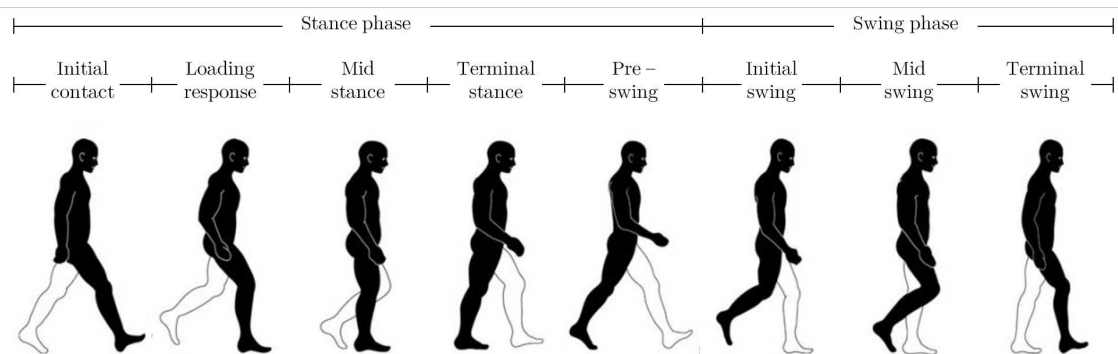


Figure 2.1: Phases and divisions of the gait cycle according to [12]. Adapted from [11].

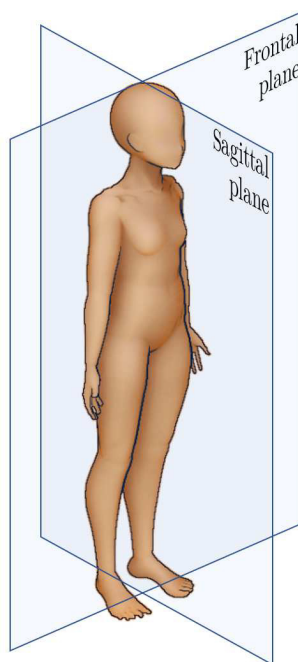


Figure 2.2: Anatomical position and vertical planes of the human body.

2.1.1.1 HIP JOINT

FLEXION AND EXTENSION. These movements consist of moving the leg forth and back to the anatomical position in the sagittal plane, as seen in Figure 2.3a, respectively.

HYPEREXTENSION. This movement consist of moving the leg farther posteriorly the anatomical position in the sagittal plane, as shown in Figure 2.3b.

ABDUCTION AND ADDUCTION. These movements consist of moving the leg outward and inward the anatomical position in the frontal plane, respectively. These movements are depicted in Figure 2.3c.

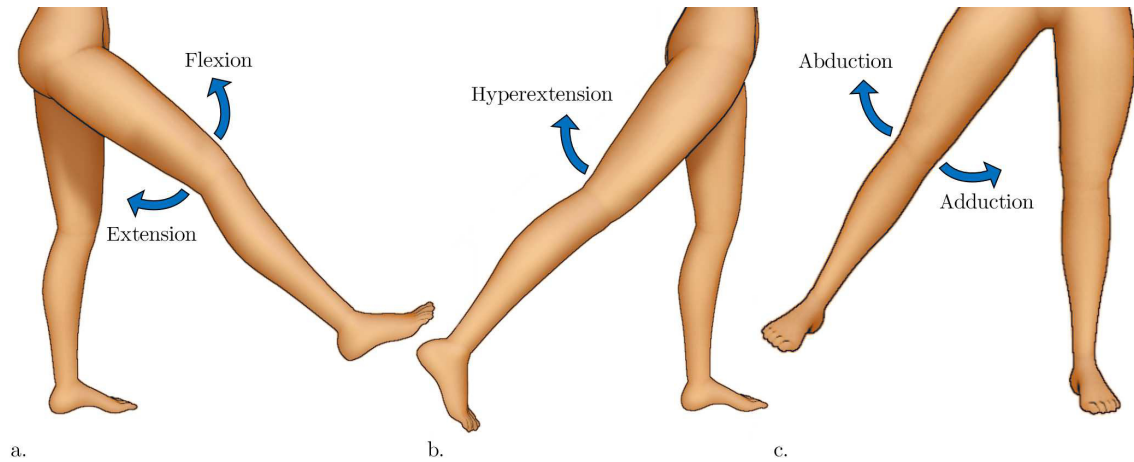


Figure 2.3: Movements of the hip joint.

2.1.1.2 KNEE JOINT

FLEXION AND EXTENSION. These movements bend and straighten the leg at the knee in the sagittal plane, as seen in Figure 2.4.

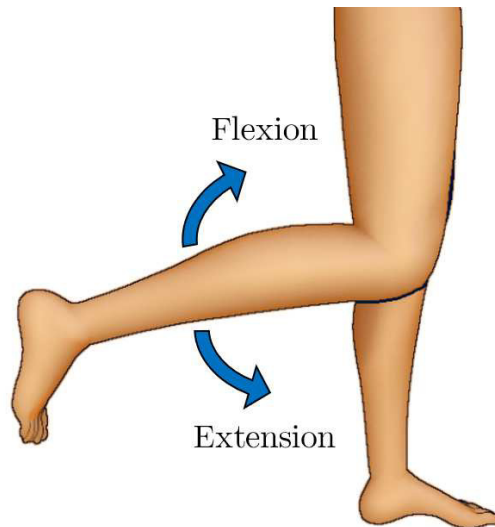


Figure 2.4: Movements of the knee joint.

2.1.1.3 ANKLE JOINT

DORSIFLEXION. This movement consist of moving the ankle so that the tip of the foot points upwards.

PLANTARFLEXION. This is the opposite movement of the dorsiflexion, so that the tip of the foot points downwards. Both previous movements are shown in Figure 2.5.

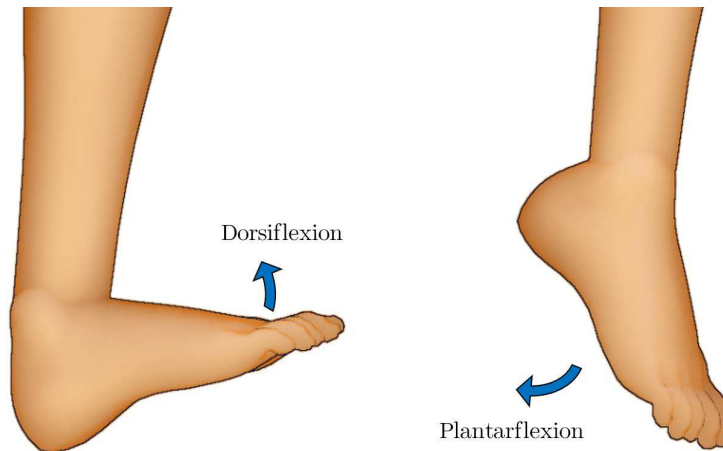


Figure 2.5: Movements of the ankle joint.

2.1.2 NEURAL SUPPLY OF THE LOWER-LIMB

Muscles have to be activated in order to perform movements, such as those of the gait cycle. The nervous system is responsible for identifying the specific muscles necessary to perform a particular movement and for generating the stimulus necessary to develop the force that is required in such muscles. The central nervous system is divided into the central nervous system and the peripheral nervous system. In the specific case of movement, the central nervous system is responsible for the initiation and control of movement and consist of the brain and the spinal cord; and the peripheral system is responsible of muscle activation and consist of spinal nerves.

As the main control is carried out in the brain, the acquisition of brain activity is of the utmost importance. Such acquisition can be carried out by a wide number of techniques, where the most applied are the ECoG and the EEG. Since the EEG is a non-invasive method, as it does not require a surgical procedure (compared to

the ECoG), it is preferred for performing measurements of brain electrical activity.

2.1.2.1 EEG

As mentioned earlier, EEG is a non-invasive method for the continuous measurement of electrical activity in the brain in which electrical conductors, known as electrodes, are placed along the scalp. The number and placement of electrodes vary according to the standard being used. For example, Figure 2.6 shows the electrode placement specified by the international 10-20 system [13], which is used for most applications. Each electrode has a letter to identify the region of the brain it is measuring from, such as pre-frontal (Fp), frontal (F), temporal (T), central (C), parietal (P) and occipital (O). The amplitudes of the EEG signals for each electrode ranges from 10 to 100 μV .

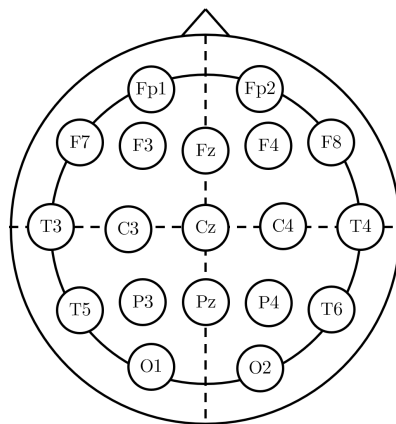


Figure 2.6: A single plane projection of the head with electrode placement according to the international 10-20 system.

FREQUENCY BANDS OF THE EEG. The EEG has a fuzzy frequency range, although it can be broken down into the following bands or rhythms [14]:

DELTA (δ). Below 3.5 Hz, these frequencies are related to sleep and anesthesia, and they are often used to characterize the depth of sleep or type of anesthesia.

THETA (θ). This band occur within the 4 to 7.5 Hz frequency range, and is related to motion, alert state and cognitive tasks.

ALPHA (α). Mainly observed in the frequency range of 8 to 13 Hz, this band is associated with relaxation states. A variant of this band is the Mu (μ) band, this is observed in the motor cortex around the same frequency range.

BETA (β). Ranging from 14 to 30 Hz, this band is observed as the slow bands disappear during wakefulness.

GAMMA (γ). Above 30 hz, these frequencies are enhanced during cognitive tasks, arousal and stimulation.

BEREITSCHAFTSPOTENTIAL (BP). The BP is the pre-motor potential measured in the motor cortex during EEG recordings that precedes voluntary movement in humans. It is divided into two segments: the initial slow segment (early BP) and the steeper negative slope (late BP), where the first begins about 2 seconds before the movement and the latter begins about 400 milliseconds before the movement [15].

ARTIFACTS. The recordings of EEG are subject to interference known as artifacts. These can be derived from a variety of sources that hide the original signal, such as blinks, eye movements, heart beats, muscle noise and line noise. Improper handling of those artifacts can lead to misleading interpretations of the processes studied with such EEG.

Several methods for its identification and posterior attenuation of artifacts have been developed. One of the simplest yet effective methods for the removal of artifacts is the Blind Source Separation (BSS) [16]. The main idea of the BSS method is to recover independent sources $\mathbf{s}(k) = s_1(k), \dots, s_N(k)$ after being mixed by an unknown matrix M to a linear mix of signals $\mathbf{x}(k) = x_1(k), \dots, x_N(k)$. The recovered version $\mathbf{r}(k)$ of the original sources $\mathbf{s}(k)$ is computed with

$$\mathbf{r}(k) = \mathbf{W}\mathbf{x}(k) , \tag{2.1}$$

where \mathbf{W} is the recovery matrix consisting of filters that attempt to invert the mixing process.

2.2 MACHINE LEARNING (ML) CONTEXT

In this section, concepts and algorithms belonging to the ML framework, such as times series matching, clustering, regression and optimization, are explained. For evaluating these methods, different measures and metrics are also defined.

2.2.1 TIME SERIES MATCHING

In the field of time series analysis, similarity measures between time series is one of the most prevalent problems for finding sections of signals that match in some characteristic. For finding similar patterns between series, several methods of time series matching have been proposed. If the characteristic to be found is the similarity in shape, regardless of the amplitude and the size of the signals, the dynamic time warping is one of the most common and reliable choices.

2.2.1.1 DYNAMIC TIME WARPING (DTW)

The DTW is an algorithm used to align time series in the search of shape similarity by creating a warping path that minimizes the distance between elements of the series. The main advantages of DTW are that time series can be aligned even if they are dephased one from another, and that time series are not required to be within the same range of magnitude. The band-constrained DTW, a slight modification that improves the resulting path [17], is explained as follows.

Two time series, a sequence \mathbf{a} of length I and a sequence \mathbf{b} of length J , are expressed as:

$$\begin{aligned}\mathbf{a} &= \{a_1, a_2, a_3, \dots, a_I\} \quad \text{and} \\ \mathbf{b} &= \{b_1, b_2, b_3, \dots, b_J\} .\end{aligned}\tag{2.2}$$

Then a distance matrix $\mathbf{D}_{I \times J}$ between both time series is filled by computing the absolute distance d_{ij} between every pair of elements a_i and b_j in both time series:

$$d_{ij} = |a_i - b_j| ,\tag{2.3}$$

where $i = 1, \dots, I$ and $j = 1, \dots, J$.

For creating the warping path, an auxiliary matrix $\mathbf{U}_{K \times L}$ is initialized with symbolic representations of positive infinite values (or with values of very large magnitude compared to the magnitude of both original series), where $K = I + 1$ and $L = J + 1$. The value of $u_{1,1}$ is initialized as 0 and $u_{2,2}$ is initialized as $d_{1,1}$. Then, a bound constraint for the warp search is defined as:

$$W_c = \left\lceil \frac{b_p \times J}{100} \right\rceil, \quad (2.4)$$

where $\lceil \cdot \rceil$ is the ceil function and b_p is the maximum percentage of the total length of the shortest series that the warp can extend to the left (LW_c) and to the right (RW_c). Then \mathbf{U} is filled recursively in the permissible extension given by W_c with:

$$u_{ij} = d_{(i-1,j-1)} + \min(u_{(i-1,j-1)}, u_{(i-1,j)}, u_{(i,j-1)}). \quad (2.5)$$

The path with minimum distance from $u_{1,1}$ to $u_{K,L}$ is the resulting warping path, whose indexes are used to compute the sequences that map \mathbf{a} to \mathbf{b} (\mathbf{S}_{ab}) and \mathbf{b} to \mathbf{a} (\mathbf{S}_{ba}). An example result of the DTW and the effect of the bound constraint in the warping path is seen in Figure 2.7. Algorithm 2.1 summarizes the mapping with DTW.

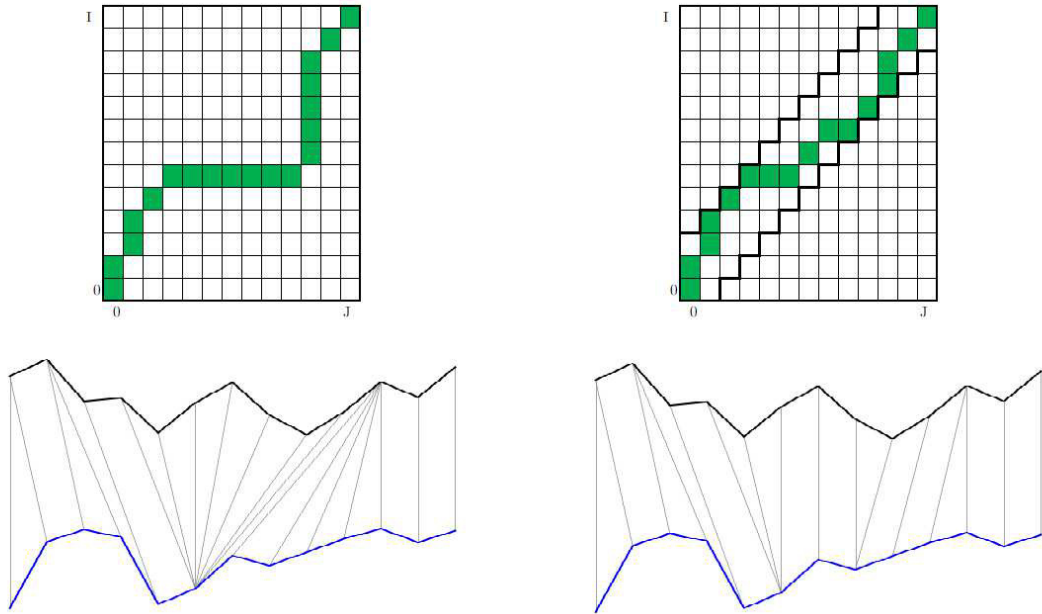


Figure 2.7: Effect of the bound constraint in the generation of the warping path in the DTW algorithm.

```

1: procedure DTW( $\mathbf{a}$ ,  $\mathbf{b}$ ,  $b_p$ )
2:    $I \leftarrow$  length of  $\mathbf{a}$ 
3:    $J \leftarrow$  length of  $\mathbf{b}$ 
4:   Distance matrix:
5:   Define  $\mathbf{D} \leftarrow$  empty  $I \times J$  matrix
6:   for  $i$  from 1 to  $I$  do
7:     for  $j$  from 1 to  $J$  do
8:        $d_{ij} \in \mathbf{D} \leftarrow$  computation of Equation 2.3
9:   Auxiliary matrix:
10:   $K \leftarrow I + 1$ 
11:   $L \leftarrow J + 1$ 
12:  Define  $\mathbf{U} \leftarrow K \times L$  matrix filled with  $\infty$ 
13:   $u_{1,1} \in \mathbf{U} \leftarrow 0$ ,  $u_{2,2} \in \mathbf{U} \leftarrow d_{1,1}$ 
14:   $W_c \leftarrow$  computation of Equation 2.4
15:  for  $i$  from 2 to  $K$  do
16:     $LW_c \leftarrow \max(2, i - \lceil W_c K / 2 \rceil)$ 
17:     $RW_c \leftarrow \max(2, i + \lceil W_c K / 2 \rceil)$ 
18:    for  $i$  from  $LW_c$  to  $RW_c$  do
19:       $u_{ij} \leftarrow$  computation of Equation 2.5
20:  Warping path:
21:   $i \leftarrow I$ 
22:   $j \leftarrow J$ 
23:   $\mathbf{S}_{ab} \leftarrow i - 1$ 
24:   $\mathbf{S}_{ba} \leftarrow j - 1$ 
25:  while  $i = j \neq 2$  do
26:     $min_{ind} \leftarrow \min(u_{(i-1,j-1)}, u_{(i-1,j)}, u_{(i,j-1)})$ 
27:    if  $u_{(i-1,j-1)} = \min(u_{(i-1,j-1)}, u_{(i-1,j)}, u_{(i,j-1)})$  then
28:       $i \leftarrow i - 1$ 
29:       $j \leftarrow j - 1$ 
30:    else if  $u_{(i-1,j)} = \min(u_{(i-1,j-1)}, u_{(i-1,j)}, u_{(i,j-1)})$  then
31:       $i \leftarrow i - 1$ 
32:    else
33:       $j \leftarrow j - 1$ 
34:     $\mathbf{S}_{ab} \leftarrow i - 1$ 
35:     $\mathbf{S}_{ba} \leftarrow j - 1$ 
36:  Flip  $\mathbf{S}_{ab}$ 
37:  Flip  $\mathbf{S}_{ba}$ 
   return  $\mathbf{S}_{ab}$  and  $\mathbf{S}_{ba}$ 

```

Algorithm 2.1. Pseudocode of DTW

2.2.2 CLUSTERING

Clustering methods consist of grouping points of a set so that points in the same group are as similar as possible and points of different groups are as different as possible. In this work, the similarity is the distance between elements. The results of most of the clustering methods are dependent of the number of clusters defined prior the execution of the algorithms. This can be a drawback, since most of the datasets have underlying structures that cannot be easily recognized, thus the number of clusters in the dataset are often unknown. The drawbacks of assigning an arbitrary number of clusters are depicted in Figure 2.8; where the optimal clustering may not be reached. Hence the need to use algorithms that do not require the pre-specification of the number of clusters.

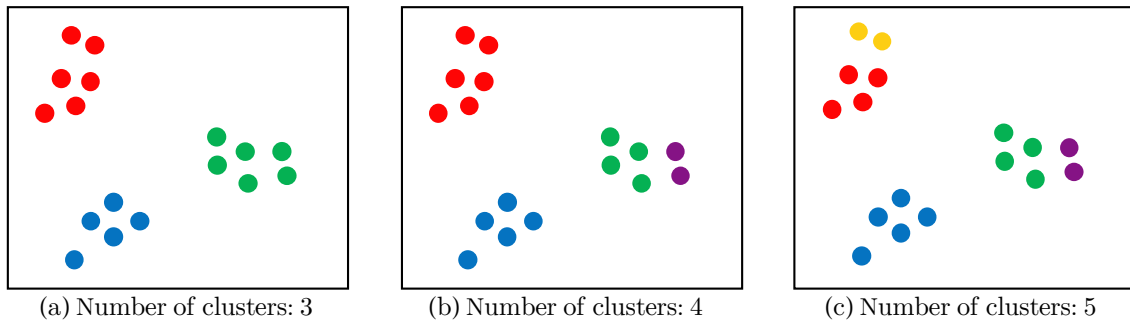


Figure 2.8: Effect of the arbitrary assignation of number of clusters. (a) When the number of clusters is 3, the optimal clustering is reached. (b) and (c) shows a deficient clustering, consequence of an arbitrary assignment of the number of clusters.

2.2.2.1 BALANCED ITERATIVE REDUCING AND CLUSTERING (BIRCH)

BIRCH is a clustering algorithm that finds good clusters with a single scan of the dataset and does not need the number of clusters as parameter. It is based on the hierarchization of the data set, which is done by building a tree data structure. The main parts of the algorithm are explained as follows [18].

CLUSTERING FEATURE (CF). A CF is a triple representing a cluster \mathbf{x} of N points, it is defined as:

$$\mathbf{cf} = (N, LS, SS), \quad (2.6)$$

where N is the number of points in the cluster, LS is the sum of points in \mathbf{x} ($\sum_{i=1}^N x_i$),

and SS is the square sum of points in \mathbf{x} ($\sum_{i=1}^N x_i$). CFs have additive property, which means that a CF could be composed of other CFs:

$$\mathbf{cf}^{(1-2)} = \mathbf{cf}^{(1)} + \mathbf{cf}^{(2)} = (N^{(1)} + N^{(2)}, LS^{(1)} + LS^{(2)}, SS^{(1)} + SS^{(2)}) . \quad (2.7)$$

So CFs of clusters are computed as clusters are merged in the tree structure.

CF-TREE. A CF-tree is a compact representation of the dataset with two parameters: branching factor B and threshold T . Each nonleaf node is a cluster containing at most B leaf nodes or subclusters with at most L elements each satisfying T , a factor of the leaf node's diameter, as seen in the Figure 2.9.

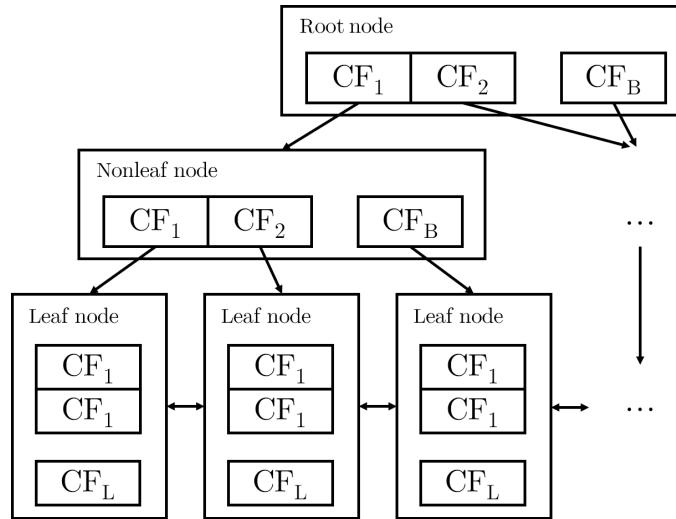


Figure 2.9: Structure of a CF-tree.

The CF-tree is built as new elements l are inserted by following the next steps:

1. Choose the leaf node with mean c that is closest to l according to:

$$d = |c - l| . \quad (2.8)$$

2. Update the leaf node if l can be entered without violating the threshold condition. Otherwise, insert a new CF element in the leaf node. When the leaf node has reached its maximum number of elements L , the node is split by choosing the farthest pair of elements and dividing in between them.

3. Modify the path to the leaf node by updating \mathbf{cf} of each nonleaf element on the path to the leaf node. If the nonleaf node has reached B elements, a split similar to step 2 must be done.

At the end of the scan, the number of leaf nodes or subclusters becomes the estimated number of clusters C for the dataset and the leaf nodes are returned as the final clusters $\Lambda = \{\lambda_1, \lambda_2, \dots, \lambda_C\}$. The summary of BIRCH is presented in Algorithm 2.2.

```

1: procedure BIRCH( $data, T$ )
2:    $B \leftarrow$  length of  $data$  if number of clusters is unknown
3:    $L \leftarrow$  length of  $data$ 
4:   CF-tree:
5:   for  $l \in data$  do
6:     Select closest leaf node to  $l$  with Equation 2.8
7:     if  $T$  condition is met then
8:       Insert  $l$  to leaf node
9:       Update every  $\mathbf{cf}$  triplet
10:    else
11:      if Not  $L$  elements in leaf node (or  $B$  in nonleaf node) then
12:        Insert  $l$  as a single cluster
13:        Update every  $\mathbf{cf}$  triplet
14:      else
15:        Split leaf node to create another leaf node
16:        Create a new CF for the new leaf node
17:        Update every  $\mathbf{cf}$  triplet
18:   Final clusters:
19:   return leaf nodes as clusters  $\Lambda$ 

```

Algorithm 2.2. Pseudocode of BIRCH

2.2.2.2 CONSISTENCY OF CLUSTER ASSIGNMENT

The validation of the assignment of clusters can be graphically evaluated with the silhouette algorithm [19]. This algorithm assigns coefficients (silhouette coefficients) to each sample of each cluster. Such coefficients lies in the range of -1 to 1. Silhouette coefficients near to +1 indicate that the sample is assigned to a very representative cluster, coefficients near to 0 indicate that the sample is between two clusters, and coefficients near to -1 indicates that the sample might be assigned to the wrong cluster. Such coefficients are defined as follows.

Let i be a point in its cluster $\lambda^{(i)}$. For each point i , the mean distance a_i of i to every other point in $\lambda^{(i)}$ is defined as

$$a_i = \frac{1}{\|\lambda^{(i)}\| - 1} \sum_{j \in \lambda^{(i)}, i \neq j} d_{i,j} , \quad (2.9)$$

where $d_{i,j}$ is the distance between points i and j . The minimal mean distance b_i of i to all points in another cluster λ is

$$b_i = \min \left(\frac{1}{\|\lambda\|} \sum_{j \in \lambda} d_{i,j} \right) \quad \forall \lambda \in \Lambda \mid \lambda \neq \lambda^{(i)} , \quad (2.10)$$

The cluster λ with the smallest b_i is the neighbor cluster of i . Then the silhouette value s_i for each point i is computed as follows:

$$s_i = \begin{cases} 1 - a_i/b_i, & \text{if } a_i \leq b_i , \\ 0, & \text{if } a_i = b_i , \\ b_i/a_i - 1, & \text{if } a_i \geq b_i . \end{cases} \quad (2.11)$$

However, for a better evaluation of the cluster assignment, multiple executions of the silhouette algorithm might be needed to determine if the assignment is consistent along the executions. The analysis of silhouette coefficients in Figure 2.10a depicts the quality of the cluster assignment shown in Figure 2.10b.

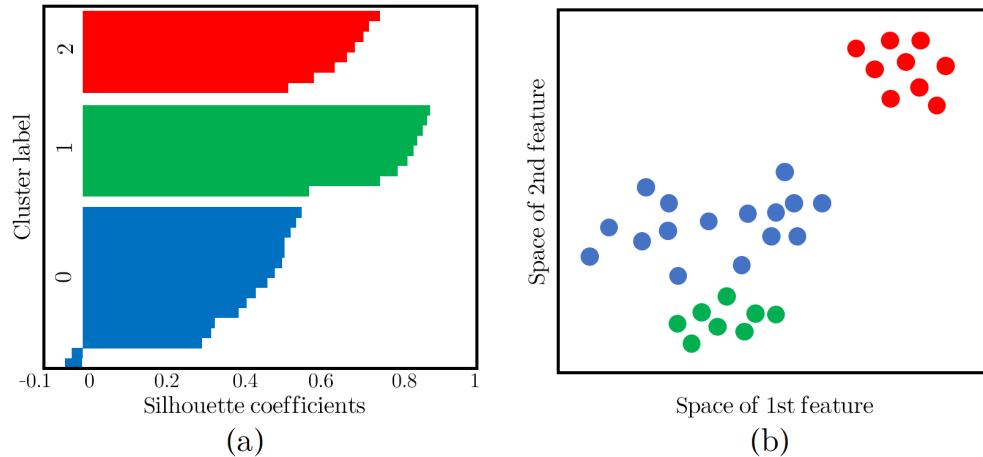


Figure 2.10: Example of a silhouette analysis on sample data with three clusters. (a) Shows the silhouette plot of the cluster assignment, two of the values of cluster 0 (*blue dots and blue bars*) might be wrongly assigned. (b) A visualization of the clustered data.

2.2.2.3 DENSITY OF CLUSTERS

One or more clusters are assigned as a result of the clustering algorithm. Each of the clusters contains a defined number of elements that are scattered or concentrated in the space of the dataset. When executing regression tasks with similarity measures (see subsection 2.2.3), one may expect several clusters if the dataset is suspected to be noisy even if the subset of points should be ideally a single cluster. In this case, computing the density of the resulting clusters is useful for determining the cluster that better represents the dataset. This can be achieved by using a Kernel Density Estimator (KDE) over the N observations, that is, by weighting the distances of the observations x_i from a particular point x . This is expressed as:

$$\hat{f}^{kernel}(x) = \frac{1}{Nb} \sum_{i=1}^N \Gamma\left(\frac{x - x_i}{b}\right), \quad (2.12)$$

where b is the width of the bin containing x and $\Gamma(\cdot)$ is the Kernel, which is the Gaussian function of a random variable x with expected value μ and variance σ^2 :

$$\Gamma(x, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\left(\frac{x-\mu}{2\sigma}\right)^2}. \quad (2.13)$$

The bin with the largest density estimation is the densest cluster.

2.2.3 REGRESSION

The regression problem is one in which a continuous variable is estimated from a set of inputs. Some of the most widely used methods for solving regression tasks are the k -nearest neighbors and the multi-layer perceptron algorithms, which are described below.

2.2.3.1 K-NEAREST NEIGHBORS (KNN)

The K nearest vectors or neighbors of a query vector \mathbf{x} , contained in a dataset \mathbf{S} , are obtained with a similarity measure function f_s as in Algorithm 2.3. Then, the KNN algorithm for regression [20] estimates the output y of a query vector \mathbf{x} as the mean of the outputs o of its K nearest neighbors in \mathbf{S} , as in the following equation:

$$y(\mathbf{x}) = \frac{1}{K} \sum_{i=1}^K o_i(\mathbf{x}). \quad (2.14)$$

```

1: procedure KNN_SEARCH( $\mathbf{x}$ ,  $\mathbf{S}$ ,  $f_s$ ,  $K$ )
2:   Initialize empty lists  $\mathbf{D}$  and  $\mathbf{I}$ 
3:    $N \leftarrow$  length of  $\mathbf{S}$ 
4:   for  $i$  from 1 to  $N$  do
5:     Append  $f_s(x, S_i)$  to  $\mathbf{D}$ 
6:     Append  $i$  to  $\mathbf{I}$ 
7:   Arrange  $\mathbf{I}$  according to  $\mathbf{D}$  sorted in descending order.
8:    $\boldsymbol{\kappa} \leftarrow$  first  $K$  elements of  $\mathbf{I}$ 
9:   return  $\boldsymbol{\kappa}$  ▷ Indexes of the  $K$  nearest neighbors

```

Algorithm 2.3. Pseudocode of KNN search.

This means that all of the K nearest neighbors of \mathbf{x} contribute equally to the estimation of its output. Another approach used for the output estimation is to weight the output o of each one of the K nearest neighbors [21], this weighting is done as follows:

$$y(\mathbf{x}) = \frac{1}{K} \sum_{i=1}^K \frac{1}{f_s(\mathbf{x}, \mathbf{S}_i)} o_i(\mathbf{x}), \quad (2.15)$$

where f_s is usually the euclidean distance.

2.2.3.2 MULTI-LAYER PERCEPTRON (MLP)

The architecture of the MLP is shown in Figure 2.11. This artificial neural network consist of one input layer, multiple hidden layers, and one output layer.

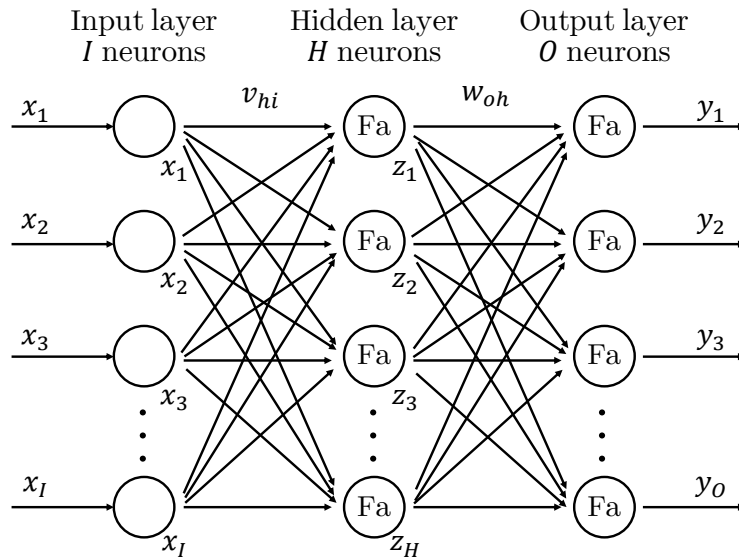


Figure 2.11: Architecture of a MLP.

The number of neurons in the input, hidden and output layers are I , H , and O , respectively. The outputs of each neuron of the input layer are the same as its inputs $x_i \in \mathbf{x}$, since this layer is composed of identity neurons. Each neuron in the hidden layer takes the weighted sum of the outputs of the input layer, such that the output of these neurons z_i is expressed as:

$$z_h = Fa \left(\sum_{i=1}^H v_{hi} x_i + b \right) , \quad (2.16)$$

where $v_{hi} \in \mathbf{v}$ is the weight of neuron i of the input layer to neuron h of the hidden layer, b is a bias term and Fa is the activation function that introduces non-linearity to MLP. The most commonly used Fa is the sigmoidal function, which is expressed as:

$$Fa(\mathbf{x}) = \frac{1}{1 + e^{-\mathbf{x}}} . \quad (2.17)$$

The output of each neuron y_o in the output layer is defined similarly to the hidden layer with:

$$y_o = Fa \left(\sum_{h=1}^O w_{oh} z_h + b \right) , \quad (2.18)$$

where $w_{oh} \in \mathbf{w}$ is the weight of neuron h of the hidden layer to neuron o of the output layer. Weights \mathbf{v} and \mathbf{w} must be optimized to decrease the following loss function:

$$loss = \frac{1}{N} \sum_{i=1}^N (y_{o_i} - y_{d_i})^2 , \quad (2.19)$$

where N is the length of the dataset and y_{o_i} and y_{d_i} are the estimated and desired outputs, respectively. One of the most efficient learning algorithms for the optimization of such weights is the adaptive moment estimation method [22].

ADAPTIVE MOMENT ESTIMATION (ADAM). One of the most attractive benefits of using the Adam method is the intuitive interpretation and little tuning of the hyper parameters. The main idea of the Adam method is to compute individual adaptive learning rates for different parameters θ of a loss function from estimations of first and second moments of gradients g . The first moment of the gradient (the

decaying average of previous gradients) m_k and the second moment of the gradient (previous squared gradients) v_k at a time step k are computed as follows:

$$\begin{aligned} m_k &= \beta_1 m_{k-1} + (1 - \beta_1) g_k , \\ v_k &= \beta_2 v_{k-1} + (1 - \beta_2) g_k^2 , \end{aligned} \quad (2.20)$$

where β_1 and β_2 are exponential decay rates for the moment estimates. Then, the bias-corrected first and second moments are computed with:

$$\begin{aligned} \hat{m}_k &= \frac{m_k}{1 - \beta_1^k} , \\ \hat{v}_k &= \frac{v_k}{1 - \beta_2^k} . \end{aligned} \quad (2.21)$$

Finally, the learning rates for the different parameters are updated according the following rule:

$$\theta_{k+1} = \theta_k - \frac{\alpha}{\sqrt{\hat{v}_k + \epsilon}} \hat{m}_k , \quad (2.22)$$

where α is the learning rate (the size of the steps made for reaching a minimum), ϵ is a term used to avoid division by zero, and θ_k are the parameters of the loss function at a time step k . In the previous equations β_1 and β_2 are usually set as 0.99, and ϵ is set as 10^{-8} , as recommended in [22]. Algorithm 2.4 summarizes the MLP neural network using Adam for optimizing the weights.

2.2.3.3 MEASURES OF GOODNESS OF FIT

Some useful measures for the evaluation of regression models are defined in this section. The goodness of fit is computed with respect to desired (\mathbf{y}_d) and estimated outputs (\mathbf{y}_e) of length N .

COEFFICIENT OF CORRELATION (R). R is a measure of the linear relationship between two variables. The following expression of R gives values from -1 to 1:

$$R = \frac{\sum_{i=1}^N (\mathbf{y}_{e_i} - \bar{\mathbf{y}}_e)(\mathbf{y}_{d_i} - \bar{\mathbf{y}}_d)}{\sqrt{\sum_{i=1}^N (\mathbf{y}_{e_i} - \bar{\mathbf{y}}_e)^2} \sqrt{\sum_{i=1}^N (\mathbf{y}_{d_i} - \bar{\mathbf{y}}_d)^2}} . \quad (2.23)$$

```

1: procedure MLP_TRAIN(train_data)
2:    $\mathbf{x} \leftarrow$  input data in train_data
3:    $\mathbf{y}_d \leftarrow$  output data in train_data
4:   Initialize random weights  $\mathbf{v}, \mathbf{w}$ 
5:   Initialize time step  $k \leftarrow 0$ 
6:   while loss has not converged do
7:      $z_h \leftarrow$  computation of Equation 2.16
8:      $\mathbf{y}_e \leftarrow$  computation of Equation 2.25
9:     loss  $\leftarrow$  computation of Equation 2.19 with  $\mathbf{y}_e$  and  $y_d$ 
10:     $k \leftarrow k + 1$ 
11:    Update  $\mathbf{v}$  and  $\mathbf{w}$  with ADAM
12:  return  $\mathbf{w}, \mathbf{v}$ 
13: procedure MLP_TEST(test_data,  $\mathbf{v}, \mathbf{w}$ )
14:    $\mathbf{x} \leftarrow$  input data in test_data
15:    $\mathbf{y}_d \leftarrow$  output data in test_data
16:    $z_h \leftarrow$  computation of Equation 2.16
17:    $\mathbf{y}_e \leftarrow$  computation of Equation 2.25
18:   return Estimated output  $\mathbf{y}_e$ 
19: procedure ADAM(loss,  $m_{k-1}, v_{k-1}, \theta_k, k$ )
20:   Initialize moment estimates:  $m_0, v_0 \leftarrow [0, 0]$ 
21:   Initialize parameters  $\epsilon, \alpha, \beta_1$  and  $\beta_2$ 
22:    $g_k \leftarrow$  gradient of loss
23:   Update moment estimates  $m_k, v_k$  with Equation 2.20
24:   Create bias-corrected estimates  $\hat{m}_k, \hat{v}_k$  with Equation 2.21
25:   Update objective parameters  $\theta_{k+1}$  with Equation 2.22
26:   return objective parameters (weights  $\mathbf{v}$  and  $\mathbf{w}$ )  $\theta_{k+1}$ 

```

Algorithm 2.4. Pseudocode of the MLP with Adam optimization

Results of R near to 1 means that the variables present positive correlation, while values near to -1 means that there exist negative correlation between the variables. If R is near to 0, there is no relationship between the two variables.

COEFFICIENT OF DETERMINATION (R^2). When comparing sinusoidal-like signals, as in some periodical signals, R may be misinterpreted. For such cases, an alternative for assessing the similarity between those signals is the R^2 , which is defined as:

$$R^2 = 1 - \frac{\sum_{i=1}^N (\mathbf{y}_{d_i} - \bar{\mathbf{y}}_d)^2}{\sum_{i=1}^N (\mathbf{y}_{d_i} - \mathbf{y}_{e_i})^2}. \quad (2.24)$$

Values of R^2 near to 1 are preferred.

SIGNAL-TO-NOISE RATIO (SNR). Another measure of goodness of fit is the SNR. It is expressed in terms of variance and mean squared error:

$$SNR = 10 \log_{10} \left(\frac{Var(\mathbf{y}_e)}{MSE(\mathbf{y}_e - \mathbf{y}_d)} \right). \quad (2.25)$$

A SNR of value 0 indicates that the signal and noise are in equal proportions. SNRs below 0 mean that there is more noise than signal, and SNRs above 0 mean that there is more signal than noise [3].

2.2.4 OPTIMIZATION

The optimization consist of maximizing or minimizing a function by searching and evaluating the best available solution in a set of proposed solutions. The most used method for optimization is the genetic algorithm.

2.2.4.1 GENETIC ALGORITHM (GA)

The GA is an evolutionary algorithm inspired by the process of natural selection. It consists of generating a random population \mathbf{P} with TI individuals of N dimensions each. In the binary GA, each proposed solution is represented as an array of 0s and 1s. Evolution in the GA is an iterative process executed during TG generations, consisting of the following steps:

1. **EVALUATION.** The generated population $\mathbf{P}_{TI \times N}$ is evaluated according to a fitness function Ff . Each individual $\mathbf{x} \in \mathbf{P}$ enters as argument to the Ff to obtain a fitness value $fv \in [0, 1]$, where 0 indicates a poor evaluation of the individual and 1 indicates a good evaluation.

$$fv(\mathbf{x}) = Ff(\mathbf{x}) \quad \forall \mathbf{x} \in \mathbf{P}. \quad (2.26)$$

2. **SELECTION.** In the basic binary GA, individuals are selected through tournament, somewhat similar to the natural selection. This involves running several tournaments between pairs of individuals, which are chosen randomly from the population. The winner of each tournament is the one with the largest fitness value fv . Winners fill the selected population \mathbf{S} to be used in the next step.

3. **CROSSOVER.** The crossover in the simple binary GA combines the information of two individuals $\mathbf{s}_i, \mathbf{s}_j \in \mathbf{S}$ (parents) to generate a new individual c (child) according to a crossover probability cp . One of the most common methods of crossover for binary individuals is the uniform crossover, in which both parents are compared to a random mask $\mathbf{z} \in \mathbb{B}^N$ to generate a child, as in the following expression:

$$c_n = \begin{cases} s_{in}, & \text{if } z_n = 1, \\ s_{jn}, & \text{if } z_n = 0, \end{cases} \quad \forall n \in N. \quad (2.27)$$

MUTATION. As an optional step, mutation swaps each element of the generated child according to a mutation probability mp .

The generated individuals update the population \mathbf{P} to be re-evaluated in step 1. When the GA completes TG generations of evaluation, selection, crossover and mutation, the individual with the largest fv is selected as the solution of the optimization. The GA is summarized in Algorithm 2.5.

```

1: procedure GENETIC_ALGORITHM( $TI, TG, N, cp, mp$ )
2: Initialization:
3:   Generate random population  $\mathbf{P} \in \mathbb{B}^{TI \times N}$ 
4:   Evaluate  $\mathbf{P}$  with  $Ff$  in Equation 3.10
5: Main loop:
6:   for 1 to  $TG$  do
7:     Select individuals from  $\mathbf{P}$  through tournament selection of its  $fv$ 
8:     for  $\mathbf{s}$  in  $\mathbf{P}$  do
9:        $random_{cp} \leftarrow \text{random}[0, 1]$ 
10:      if  $cp > random_{cp}$  then
11:        Generate child with Equation 2.27
12:       $random_{mp} \leftarrow \text{random}[0, 1]$ 
13:      for  $j$  from 1 to  $N$  do
14:        if  $mp > random_{mp}$  then
15:          Mutate element  $j$  of  $\mathbf{s}$ 
16:   return Individual  $\mathbf{s}$  with the largest  $fv$ 

```

Algorithm 2.5. Main phases of the GA

PROPOSED SCHEME

The proposed scheme to estimate torques from EEG signals consists in the following: first, raw sets of data are acquired and pre-processed. Then, an optimization of EEG channels and data prototypes is carried out. Finally, a regression model is trained on such prototypes. This ML scheme is presented in Figure 3.1, whose steps are explained in this chapter.

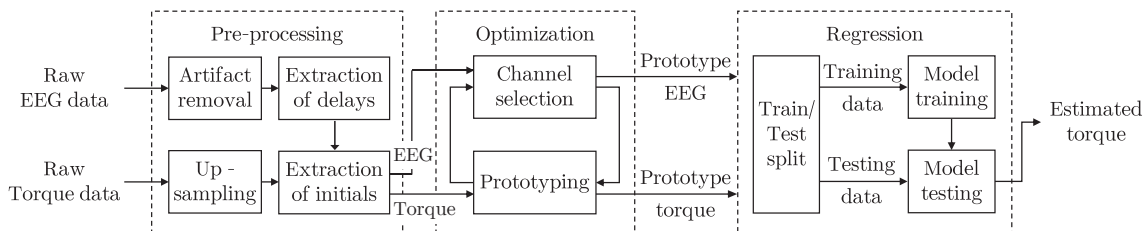


Figure 3.1: General diagram of the proposed ML scheme for torque estimation.

3.1 DATASET AND PRE-PROCESSING

A dataset containing torque and EEG signals of three gait initiation movements (tasks) executed by five subjects were acquired by the research team prior the development of this thesis. These tasks were performed at Universidad de Guadalajara by five healthy volunteers with no history of psychiatric, neurological or neurodegenerative diseases. All subjects gave their informed and signed consent to perform the tasks, which belong to an experimental protocol that was approved by the Ethics Committee of the Institute of Neurosciences of the University of Guadalajara. Each subject performed ten series, of 10 repetitions each, of the aforementioned tasks with both lower-limbs, at the time their EEG signals and

joint angles were recorded. Each series is divided in four phases: a basal state of 5 seconds, a period of 10 repetitions of a single task (time varying among subjects), a basal state of 5 seconds, and a resting period of 10 seconds. The phases of a single series is illustrated in Figure 3.2.

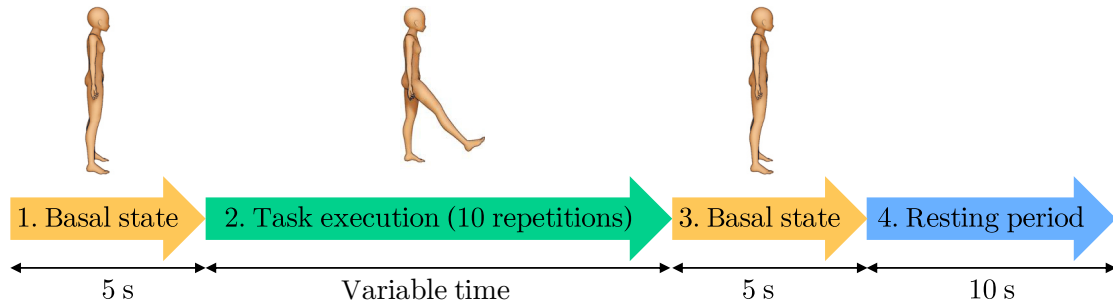


Figure 3.2: Phases of a single series of the protocol. A total of 10 continuous series of each movement is executed by each subject for each type of movement and for each lower-limb.

3.1.1 GAIT INITIATION MOVEMENTS

The gait initiation movements are observed at the beginning of forward walk, backward walk and at stair climbing. According to the definitions in subsection 2.1.1, these movements are next described from the anatomical position shown in Figure 3.3a.

MOVEMENT 1: STEP FORWARD. This movement is observed at the beginning of forward walking. It consist of a flexion of around 45° on the hip from the anatomical position, as seen in Figure 3.3b. This movement is labeled as $T1_L$ for the left lower-limb and as $T1_R$ for the right lower-limb.

MOVEMENT 2: STEP BACKWARD. A step backward is executed at the start of backward walking. It involves a hyperextension on the hip from the anatomical position. This movement is shown in Figure 3.3c. This movement is labeled as $T2_L$ for the left lower-limb and as $T2_R$ for the right lower-limb.

MOVEMENT 3: STEP UP. Observed at the beginning of stair climbing, the step up involves a flexion of around 45° on the hip and a natural extension of around 45° on

the knee, due to gravity. This can be seen in Figure 3.3d. This movement is labeled as $T3_L$ for the left lower-limb and as $T3_R$ for the right lower-limb.

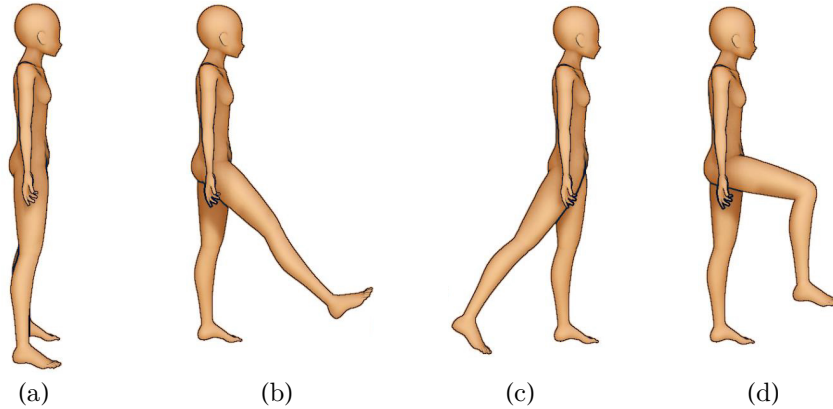


Figure 3.3: Gait initiation movements performed for the acquisition of EEG and torque data. (a) is the anatomical position, (b) is the step forward movement, (c) depicts the step backward movement and (d) shows the step up movement.

3.1.2 TORQUE ACQUISITION

The acquisition of torque was performed as follows. Markers were placed in hip, knee, ankle and toes of the lower-limb of interest for each movement and subject. The execution of each task was video recorded at a rate of 30 frames per second. By means of computer vision techniques, angles of hip, knee and ankle joints were measured. Then, kinematic and dynamical models of the lower-limb were proposed, which used as parameters the weights and heights of each subject. These models were applied to a close-loop control scheme for tracking the joint positions in order to compute the torques of each joint. Figure 3.4 shows the lower-limb kinematic chain with three degrees of freedom proposed for the measurement of angles.

UPSAMPLING OF TORQUE DATA. As both EEG and torque signals are acquired at different sampling rates, f_τ and f_{EEG} , an upsampling of the torque data is required. Given the original torque signal $\tau_O \in \mathbb{R}^M$ and its respective discrete Fourier transform $\mathbf{T}_O \in \mathbb{R}^M$, the upsampled torque τ is obtained as follows:

1. If M is odd, make $M = M - 1$
2. Zero-pad \mathbf{T}_O by inserting M zeros to the central portion of \mathbf{T}_O to create \mathbf{T}_Z . This can be done by moving the first half of \mathbf{T}_O a total of $\frac{M}{2} \left(\frac{f_{EEG}}{f_\tau} - 1 \right)$ bins

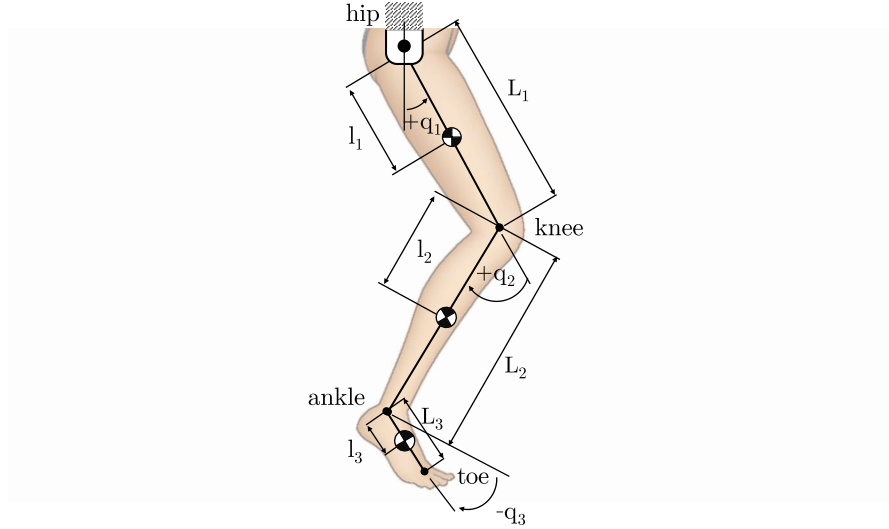


Figure 3.4: Lower-limb kinematic chain with 3 degrees of freedom, representing the hip (q_1), knee (q_2) and ankle (q_3) angles.

to the left and by moving the last half of \mathbf{T}_O a total of $\frac{M}{2} \left(\frac{f_{EEG}}{f_\tau} - 1 \right)$ bins to the right.

3. Compute the inverse discrete Fourier transform of \mathbf{T}_Z as $\boldsymbol{\tau}_Z$.
4. Scale the amplitudes of $\boldsymbol{\tau}_Z$ by $\frac{f_{EEG}}{f_\tau}$ to obtain the upsampled torque $\boldsymbol{\tau}$.

This acquisition results in a dataset of torque called $\boldsymbol{\tau} \in \mathbb{R}^M$, where M varies among subjects according to the time each one requires for performing the protocol.

3.1.3 EEG ACQUISITION

The scalp EEG signals of each subject were recorded at a rate of 200 Hz using a UPM-PLUS Grass[®] system by Natus Neurology[®], which includes the nineteen electrodes of the 10-20 system explained in subsection 2.1.2.1. The following pre-processing steps were performed.

ARTIFACT REMOVAL. The raw EEG recordings were filtered with a notch filter at 60 Hz to remove the power line interference. Then the BSS method was applied with the aim of extracting independent source signals and removing remaining artifacts.

The resulting EEG signals were validated by clinical inspection, from which it was identified that the signals contained neural activity free of artifacts.

EXTRACTION OF DELAYS. Delays of each channel of EEG are extracted to cover the information of about 0.35 seconds before each movement in the dataset, according to the late BP. Since the torque sampling rate of EEG, f_{EEG} , is larger than the sampling rate of torque, f_{τ} , the separation gap between each delayed sample is

$$\sigma = \text{round} \left(\frac{f_{EEG}}{f_{\tau}} \right). \quad (3.1)$$

Since the sampling rate of the overall dataset is f_{EEG} , the total number of samples to be covered by the delayed data is $N = 0.35f_{EEG}$. Thus, the number of delays to be extracted is

$$\delta = \frac{N}{\sigma}. \quad (3.2)$$

This acquisition results in a dataset of EEG called $\mathbf{EEG} \in \mathbb{R}^{M \times N}$, where M is the length (or number of rows) of the dataset and N is the dimension of the dataset (or number of columns), which is fixed in $N = 19(\delta + 1)$, as the number of channels is 19 and a total of δ delays were retained in addition to the actual value of each channel.

EXTRACTION OF INITIALS. As the movements of interest for this study are those of the initiation of gait, and to avoid the analysis of cycles (created by repeatedly executing each movement), only the first trial of each series was retained to create the dataset. Each first trial is called a pattern of movement. These patterns are labeled as \mathbf{P}_p from $p = 1, \dots, 10$. As the duration of the execution of each trial is different, the size of each pattern is fixed in $PS = 500$ from the end point of each trial. The process of extraction of EEG and torque data from relevant movements is shown in Figure 3.5. Thus the overall dataset \mathbf{DS} is

$$\mathbf{DS} = [\mathbf{EEG}, \boldsymbol{\tau}]_{initials} \in \mathbb{R}^{10PS \times 19(\delta+1)}. \quad (3.3)$$

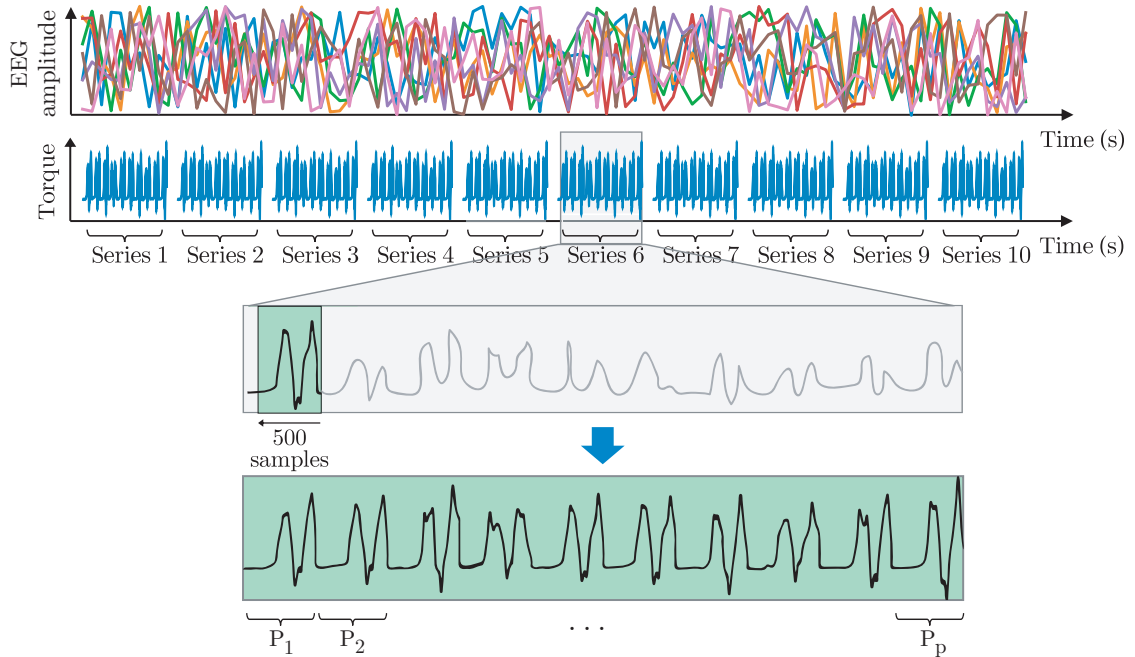


Figure 3.5: Selection of the dataset. From each one of the 10 series of trials for a task (*top figure*), only the first trial is extracted (*middle figure*). The resulting trials (*bottom figure*) are concatenated.

3.2 OPTIMIZATION STAGE

In this section, the methods for prototyping and selection of channels are described. The proposed method for prototyping the dataset is described first in this section. The motivation behind the proposal of such algorithm is to avoid the contributions of noisy data or outliers of EEG in the estimation of torque. The algorithm is based on the KNN algorithm for regression, with addition of a clustering step, which is applied assuming that the densest cluster of neighbors of a query contains its most similar points. Therefore, The algorithm attempts to validate the assumed relationship of each point of the EEG-torque dataset. The validated data, called prototype data, is optimized with a GA in order to extract the most representative dataset of the EEG-torque relationship.

3.2.1 CLUSTERIZED KNN (C-KNN) FOR PROTOTYPING

The C-KNN algorithm is used with the aim of extracting representative information of the EEG-torque relationship from the dataset. This prototyping algorithm combines time series mapping, similarity search and clustering techniques, as shown in Figure 3.6. Each step of C-KNN are next described.

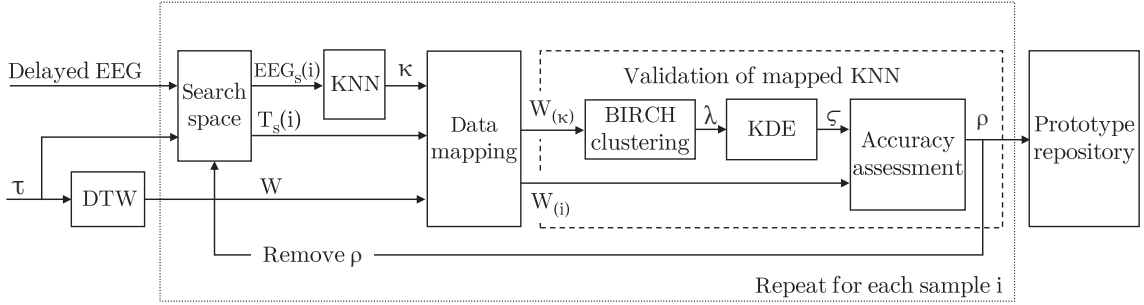


Figure 3.6: Scheme of the C-KNN prototyping algorithm.

INITIAL MAPPING. The main idea of the C-KNN algorithm is to find representative samples in the EEG-torque dataset by looking for similarity in inputs (EEG) and outputs (torque) of the dataset. This similarity measure is going to be selected in such a way that the relative proportion between input vectors (channels of the EEG) is analyzed instead of analyzing its magnitudes. Hence, the extraction of similarity in the torque data space is performed according to a shape similarity of each pattern of movement to a template movement, rather than to the magnitudes of each point in the patterns. This shape similarity is computed with the DTW algorithm, explained in subsection 2.2.1.1, as follows.

1. Each element tm of a template pattern $\mathbf{TMP} \in \mathbb{R}^{PS}$ is obtained as the normalized mean of torques of the 10 patterns extracted in subsection 3.1.3:

$$tm_n = \frac{\sum_{p=0}^{10} \mathbf{P}_{p_n}}{\max\left(\sum_{p=0}^{10} \mathbf{P}_{p_n}\right) - \min\left(\sum_{p=0}^{10} \mathbf{P}_{p_n}\right)} \quad \forall n \in PS, \quad (3.4)$$

where PS is the size of each pattern.

2. The torque of each \mathbf{P}_p is mapped to \mathbf{TMP} using the DTW algorithm, obtaining 10 warping arrays $\mathbf{w}_{\mathbf{TMP}, \mathbf{P}_p}$, which dictate the indexes of \mathbf{TMP}

to which each index of \mathbf{P}_p belongs. Each warping array is called the shape similarity of the pattern \mathbf{P}_p to the template \mathbf{TMP} .

The following steps are computed for each query point $\mathbf{i} \in \mathbf{DS}$ with index ρ .

DEFINING THE SEARCH SPACE. The space for the similarity search (\mathbf{SS}) is given by the indexes of the points of movement patterns that do not contain the query point \mathbf{i} , since points similar to the query point tend to be those that surround it. The steps for extracting the search space, based on zero-based indexing, are the following.

1. Obtain the corresponding pattern to the query point \mathbf{i} as $\mathbf{P}^{(i)} = \mathbf{P}_{\lfloor \rho/PS \rfloor}$, where PS is the size of patterns and $\lfloor \cdot \rfloor$ represents the floor operation.
2. Obtain the interval of indexes of the pattern $\mathbf{P}^{(i)}$ to be excluded with:

$$\mathbf{MOV} = [PS \lfloor \rho/PS \rfloor : PS(\lfloor \rho/PS \rfloor + 1)] . \quad (3.5)$$

3. Define the search space as:

$$\mathbf{SS} = \mathbf{DS}_{EEG} [[0 : 10PS] - \mathbf{MOV} - \mathbf{DEL}] , \quad (3.6)$$

where \mathbf{DEL} is the set of indexes of points that have been deleted from the search space (see Equation 3.8).

SIMILARITY SEARCH WITH KNN. The indexes κ of the K nearest neighbors of the EEG component of the point \mathbf{i} (\mathbf{DS}_{EEG_i}) are obtained with KNN search of Algorithm 2.3. The similarity measure used is the cosine similarity (cs):

$$cs(\mathbf{DS}_{EEG_i}, \mathbf{DS}_{EEG_j}) = \frac{\mathbf{DS}_{EEG_j} \mathbf{DS}_{EEG_i}^T}{\|\mathbf{DS}_{EEG_i}\| \|\mathbf{DS}_{EEG_j}\|} \quad \forall \mathbf{DS}_{EEG_j} \in \mathbf{SS} , \quad (3.7)$$

whose result varies from -1 to 1. The higher cs , the greater the similarity.

FINAL MAPPING. The neighbors of i are compared by mapping their torques to the template TMP using the indexes contained in $w_{TMP,P(i)}$ and by mapping the torque of the query point i (DS_{τ_i}), as seen in Figure 3.7a. This mapping forms the mapped nearest neighbors $W_{(\kappa)}$ and the mapped query point W_i shown in Figure 3.7b. Then the mapped torques are seen in terms of their normalized magnitudes, as in Figure 3.7c.

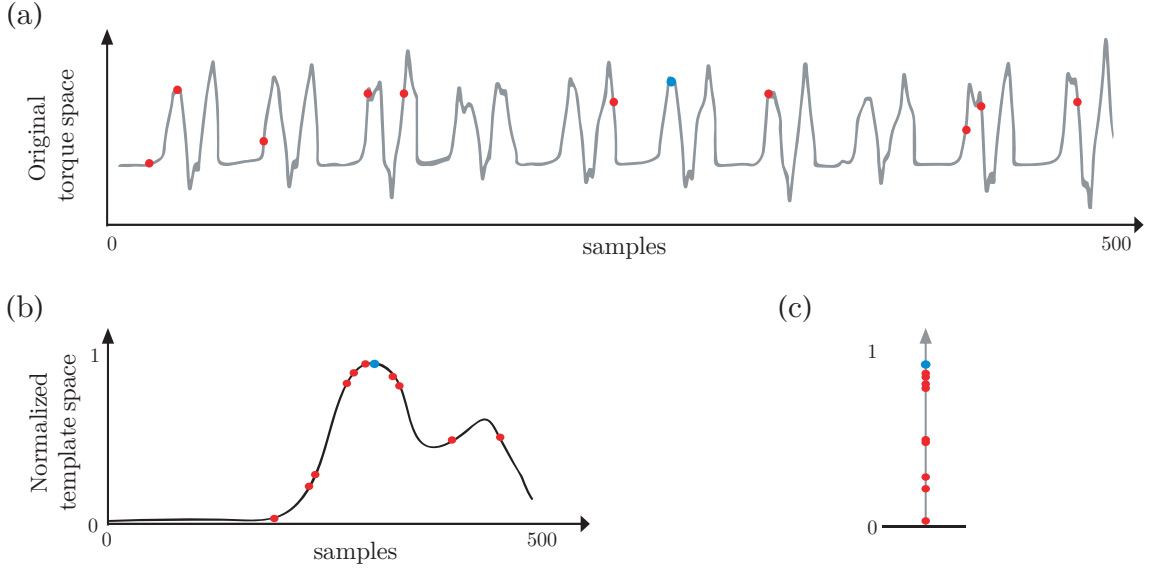


Figure 3.7: Data mapping. (a) Original torques of the K nearest neighbors (*red dots*) of a query point i (*blue dot*). (b) Mapped torques of the K nearest neighbors (*red dots*) of the mapped query (*blue dot*). (c) Magnitudes of the mapped nearest neighbors $W_{(\kappa)}$ (*red dots*) of a mapped query point W_i (*blue dot*).

VALIDATION OF MAPPED NEIGHBORS. Mapped K nearest neighbors $W_{(\kappa)}$ are clustered using the BIRCH algorithm. The resulting clusters Λ go through a Gaussian KDE to obtain the densest cluster ς , whose mean μ_{ς} is compared with $W_{(i)}$. Then the index ρ of the query point is stored as a prototype in repository PRO , or discarded as such in DEL , according to the following condition of accuracy assessment:

$$\begin{aligned} \rho &\rightarrow \mathbf{PRO}, & \text{if } W_{(i)} - \epsilon < \mu_{\varsigma} < W_{(i)} + \epsilon, \\ \rho &\rightarrow \mathbf{DEL}, & \text{otherwise;} \end{aligned} \quad (3.8)$$

where ϵ is a permissible threshold value. This validation phase is depicted in Figure 3.8.

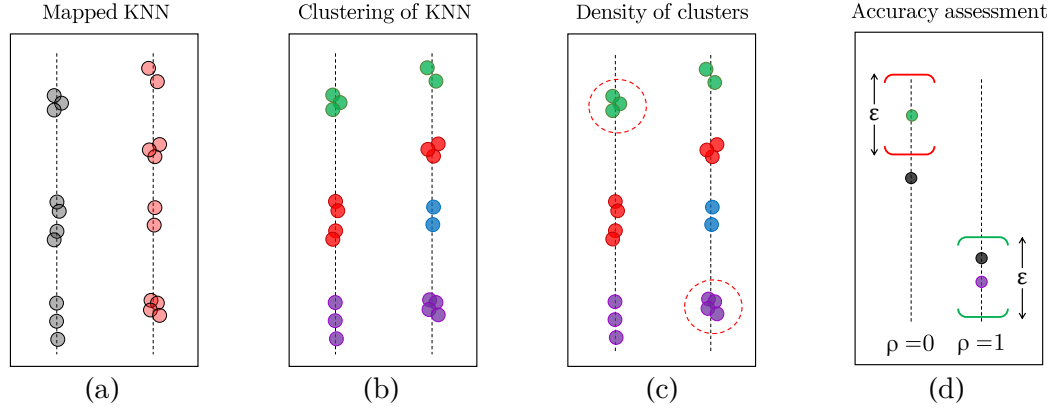


Figure 3.8: Example of the proposed validation for the KNN search. (a) shows two examples of mapped nearest neighbors $\mathbf{W}_{(\kappa)}$ (*gray and red dots*). (b) illustrates the clustering phase, where each color is a different cluster. (c) shows the densest cluster ς (*circled dots*). In (d), the mean of the densest clusters μ_{ς} (*green and purple dots*) is compared to the mapped query point $\mathbf{W}_{(i)}$ (*black dot*), ρ indicates whether the query point is stored as a prototype or not.

The summary of C-KNN is presented in Algorithm 3.1, which requires the dataset \mathbf{DS} , the size of patterns PS , the cosine similarity cs , and parameters b_p , K , ϵ and T for the DTW, KNN and BIRCH algorithms.

```

1: procedure CKNN_PROTOTYPING( $\mathbf{DS}$ ,  $b_p$ ,  $K$ ,  $\epsilon$ ,  $T$ ,  $PS$ )
2: Initial mapping:
3:    $\mathbf{TMP} \leftarrow$  computation of Equation 3.4
4:   for  $p$  from 0 to 10 do
5:      $\mathbf{W}_{P_p, \mathbf{TMP}} \leftarrow \mathbf{S}_{ab}$  of DTW( $P_p$ ,  $\mathbf{TMP}$ ,  $b_p$ ) of Algorithm 2.1
6: Main loop:
7:    $N \leftarrow$  length of  $\mathbf{DS}$ 
8:   for  $\rho$  from 1 to  $N$  do
9:      $\mathbf{i} \leftarrow \mathbf{DS}_{EEG}[\rho]$ 
10:    Define  $\mathbf{SS}$  with Equation 3.6
11:     $\kappa \leftarrow$  KNN_SEARCH( $\mathbf{i}$ ,  $\mathbf{SS}$ ,  $cs$ ,  $K$ ) of Algorithm 2.3
12: Final mapping:
13:    $\iota_0 \leftarrow$  index in  $\mathbf{W}_{P_{\lfloor \rho/PS \rfloor}, \mathbf{TMP}}$  corresponding to index  $\rho$ 
14:    $\mathbf{W}_{(i)} \leftarrow \mathbf{TMP}[\iota_0]$  ▷ Mapped query
15:    $\iota_1 \leftarrow$  indexes in  $\mathbf{W}_{P_{\lfloor \rho/PS \rfloor}, \mathbf{TMP}}$  corresponding to indexes  $\kappa$ 
16:    $\mathbf{W}_{(\kappa)} \leftarrow \mathbf{TMP}[\iota_1]$  ▷ Mapped neighbors
17: Validation of mapped KNN:
18:    $\Lambda \leftarrow$  BIRCH( $\mathbf{W}_{(\kappa)}$ ,  $T$ ) of Algorithm 2.2 ▷ Clusterize neighbors
19:    $\mu_{\varsigma} \leftarrow$  mean of  $\varsigma$  of  $\Lambda$ 
20:   Store  $\rho$  according to Equation 3.8 and  $\epsilon$ 

```

Algorithm 3.1. Pseudocode of C-KNN.

3.2.2 CHANNEL SELECTION

For each combination of subject and tasks, an optimal subset of EEG channels is selected using the GA with the aim of extracting representative cortical information. The steps carried out for the application of the GA for the purposes of this thesis are the following ones.

GENERATION OF POPULATION. A population of binary genes $\mathbf{P} \in \mathbb{B}^{TI \times 19}$ is generated. Each one of the TI individuals represents a selection of channels, with their respective delays from the EEG dataset \mathbf{DS}_{EEG} .

FITNESS FUNCTION. The EEG subset generated by each individual in \mathbf{P} is evaluated with the C-KNN prototyping algorithm, where the indexes of the resulting prototypes are used for generating the prototype datasets \mathbf{EEG}_{proto} and τ_{proto} . The GA seeks to maximize both the percentage of extracted torques τ_{proto} and their quality. By maximizing these values, the algorithm seeks for a combination of channels that could provide the largest amount of data and the most dispersed torques, so that the resulting prototypes are as similar as possible to the original torques. In this way, it is suggested that the extracted cortical information, \mathbf{EEG}_{proto} , could be more representative for the process than the original information.

The quality of the extracted data is quantified as the relative dispersion $RIQR$ between the prototype torques τ_{proto} and the original torques \mathbf{DS}_τ with the following equation:

$$RIQR = \frac{IQR_p}{IQR_o}, \quad (3.9)$$

where the interquartile ranges IQR_p and IQR_o of τ_{proto} and \mathbf{DS}_τ , respectively, are computed based on the first Q_1 and third Q_3 quartiles of the torques of interest with $IQR = Q_3 - Q_1$. Then, the fitness function F_f to be maximized is defined as follows:

$$F_{f_{\max}} = RIQR + \frac{N_p}{N_o}, \quad (3.10)$$

where N_p and N_o are the lengths of the τ_{proto} and \mathbf{DS}_τ , respectively.

The prototypes extracted with the best individuals at the end of execution of the GA ($\mathbf{DP} = [\mathbf{EEG}_{proto}, \boldsymbol{\tau}_{proto}]$) are used for estimating the torque.

3.3 TORQUE ESTIMATION

A two-layered MLP neural network was implemented for the estimation of prototype torques $\boldsymbol{\tau}_{proto}$ from prototype EEG data \mathbf{EEG}_{proto} . The extracted EEG data, from selected prototypes with the optimal subset of channels, are used as the only inputs to the neural network. The data to be estimated is the prototype torque. The estimation is made with each selection of channels proposed by the GA, according to the subject and task to which its subset of channels and optimal prototypes have been sought.

CHAPTER 4

RESULTS

In this thesis, the the main objective was to develop a ML scheme that enables the estimation of lower-limb joint torques from EEG signals acquired during gait initiation movements; such scheme was introduced in the previous chapter. In this chapter, the capability of the proposed scheme in estimating the variable of interest is assessed. Specifically, the capability of the C-KNN algorithm for obtaining prototypes and the performance of the GA for obtaining EEG channels that characterize the dataset are evaluated, so that the resulting dataset can be used in a MLP neural network to effectively estimate torques.

4.1 IMPLEMENTATION DETAILS

The parameters set to apply the main algorithms of the proposed scheme are mentioned below.

C-KNN PARAMETERS. The constraint of extension for the warping path in Equation 2.4 is fixed in $b_p = 30\%$ to avoid exaggerated warping between time series. The number of nearest neighbors to search for in the KNN algorithm is $K = 10$. The validation of neighbors found is done by fixing the permissible threshold value in Equation 3.8 as $\epsilon = 0.125$. In the BIRCH algorithm, the threshold for each node is $T = 0.5$.

GA PARAMETERS. The GA was executed 30 times, evaluating a population of $TI = 20$ individuals during $TG = 30$ generations. The number of channels the GA can select varies from 1 to $TD = 19$, where the latter is the total number of channels

or the dimension of the optimization problem. A high crossover probability $cp = 0.98$ was chosen for this study in order to exploit the solutions resulting from early generations of the GA, hence the selection of a low mutation probability $mp = 0.02$. The number of delays to be selected for each channel and the separation between them are $\delta = 10$ and $\sigma = 7$, according to the computation of equations 3.1 and 3.2 with $f_{EEG} = 200 \text{ Hz}$ and $f_{\tau} = 30 \text{ fps}$.

MLP PARAMETERS. The MLP is applied using two hidden layers with sizes $H_1 = 200$ and $H_2 = 50$. The number of neurons in the input layer is $I = \delta N_{ch}$, where N_{ch} is the number of channels selected by the GA. The output layer has a single neuron, which corresponds to the torque estimation. The bias for each neuron is set as $b = 1$ in Equation 2.16. The Adam algorithm is implemented with decay rates $\beta_1 = \beta_2 = 0.99$ in Equation 2.20, and with learning rate $\alpha = 0.125$ and $\epsilon = 10^{-8}$ in Equation 2.21, as recommended in [22]. The number of iterations the MLP was trained is 1000.

4.2 PROTOTYPES EXTRACTED

In order to evaluate the results of the prototyping stage, an analysis of the consistency of the extraction of prototypes and a comparison between prototype and original torques were performed.

4.2.1 CONSISTENCY OF EXTRACTION

The consistency of the prototyping algorithm is evaluated with a silhouette analysis of the extracted prototypes, this is done by treating the prototype torques as a cluster and the remaining data (non-prototype torques) as another cluster. The indexes of data extracted in each execution of the GA are compared with the indexes of the execution of the GA that obtained the highest fitness value in order to compute the silhouette coefficients. The plots of silhouette coefficients are shown in figures 4.1, 4.2, and 4.3 for hip, knee and ankle joints, respectively. The silhouette coefficients that lie between 0 and the mean indicates that the assignment of a sample to its cluster (prototype or non-prototype) is not as strong as the assignment of samples

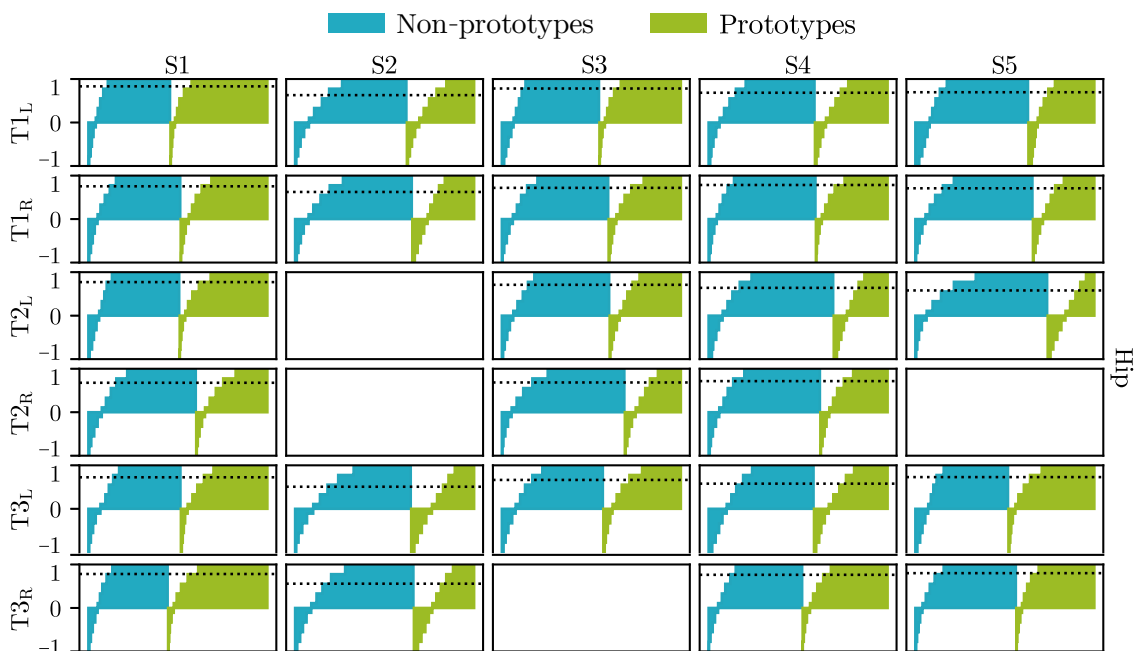


Figure 4.1: Mean (*black dotted line*) and individual silhouette coefficients of the extracted prototype (*green bars*) and non-prototype (*cyan bars*) torques after optimizing the hip data.

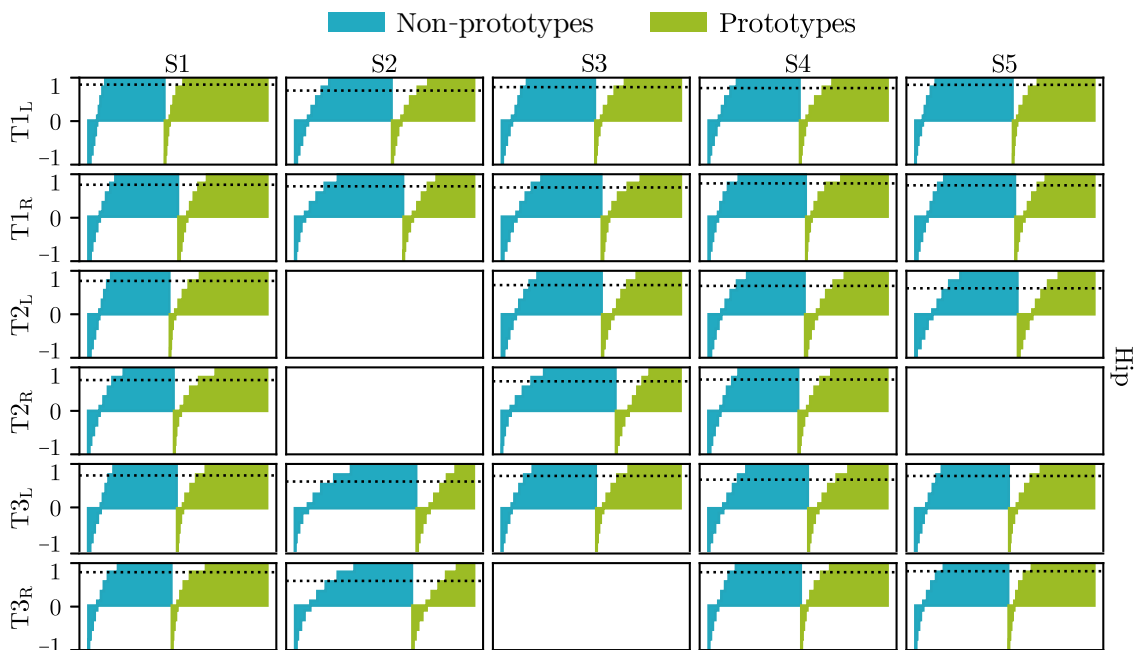


Figure 4.2: Mean (*black dotted line*) and individual silhouette coefficients of the extracted prototype (*green bars*) and non-prototype (*cyan bars*) torques after optimizing the knee data.

with values above the mean. Moreover, negative silhouette coefficients indicates that the sample may be assigned to the wrong cluster.

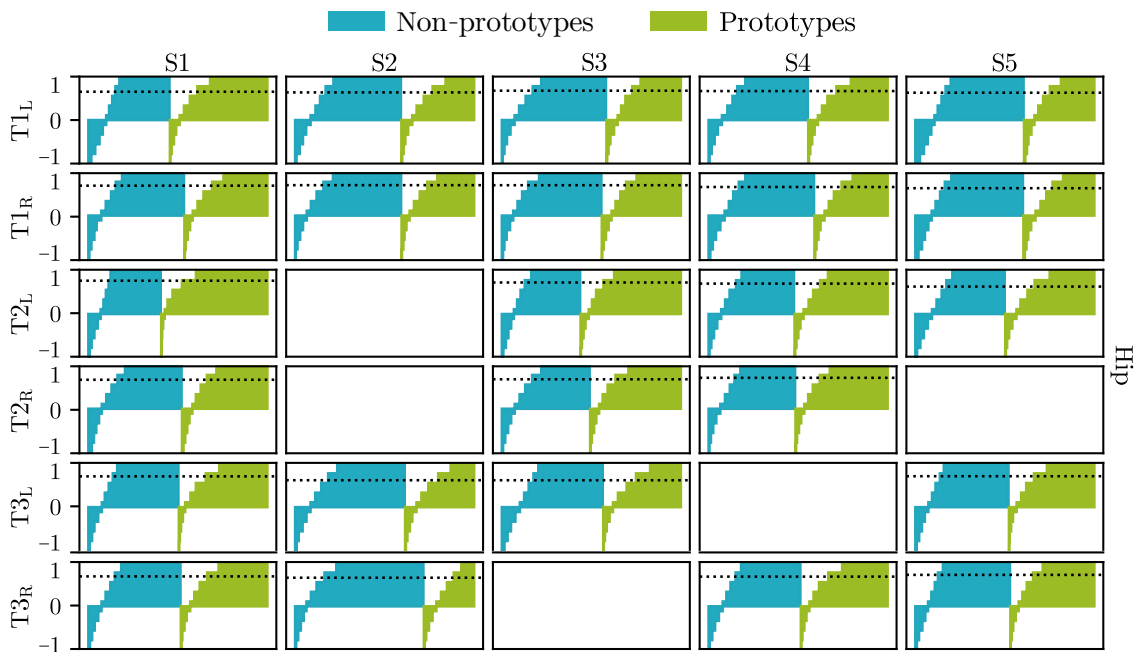


Figure 4.3: Mean (*black dotted line*) and individual silhouette coefficients of the extracted prototype (*green bars*) and non-prototype (*cyan bars*) torques after optimizing the ankle data.

4.2.2 COMPARISON TO ORIGINAL TORQUES

Prototype torques are compared to the original torques by visualizing the distributions of both datasets. Such distributions are shown in figures 4.4, 4.5 and 4.6 for hip, knee and ankle joints, respectively. Large amplitudes in the distribution curves of prototype torques around the first and third quartiles of the original torques are preferred over large amplitudes around the mean, as the latter indicates that torques were extracted mostly from data of the basal state of the movement (recall section 3.1).

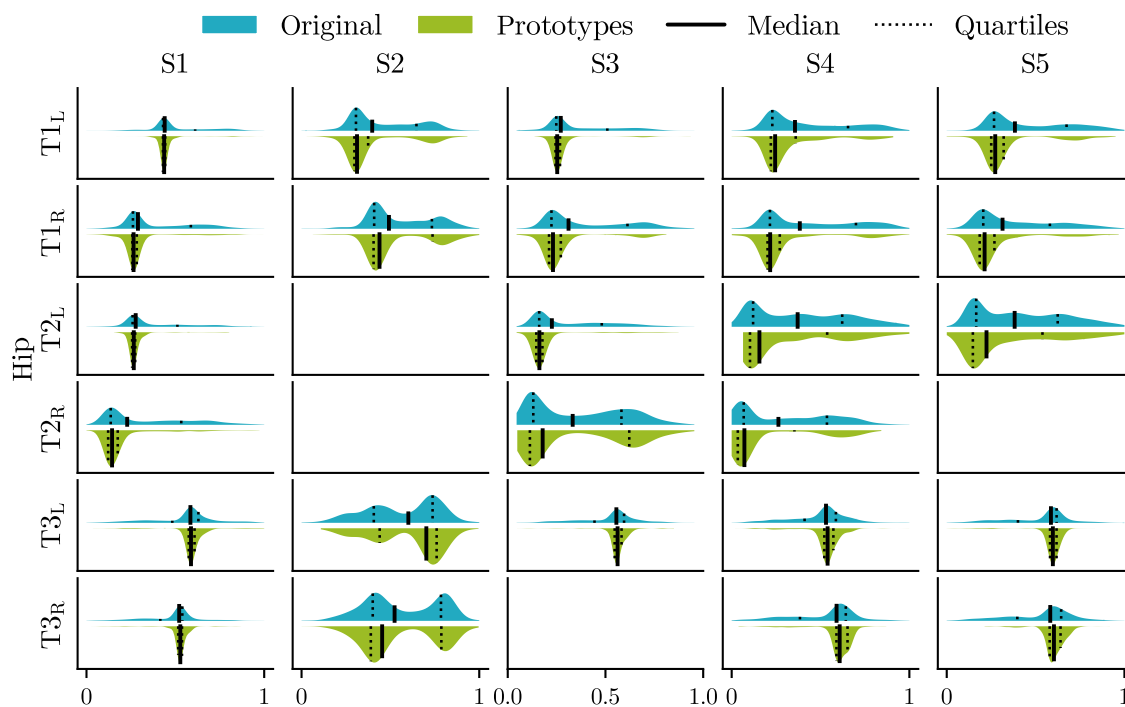


Figure 4.4: Mean (*black solid line*), 1st and 3rd quartiles (*black dotted lines*), and distributions of prototype (*green*) and original torques (*cyan*) of the hip joint.

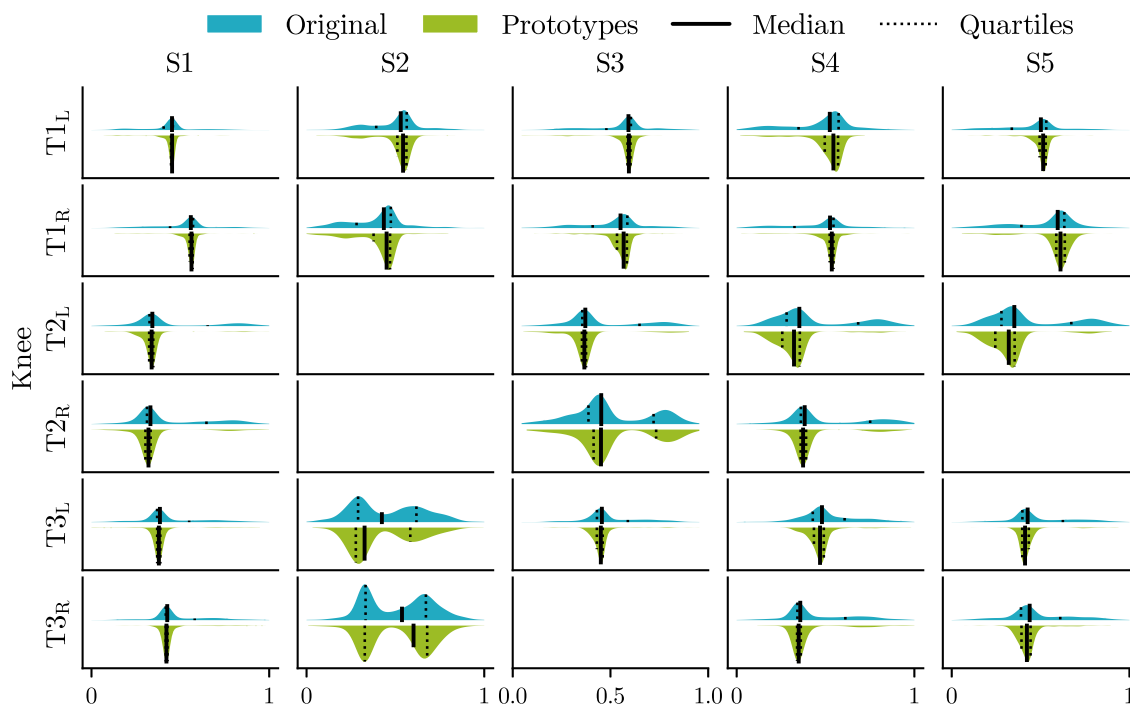


Figure 4.5: Mean (*black solid line*), 1st and 3rd quartiles (*black dotted lines*), and distributions of prototype (*green*) and original torques (*cyan*) of the knee joint.

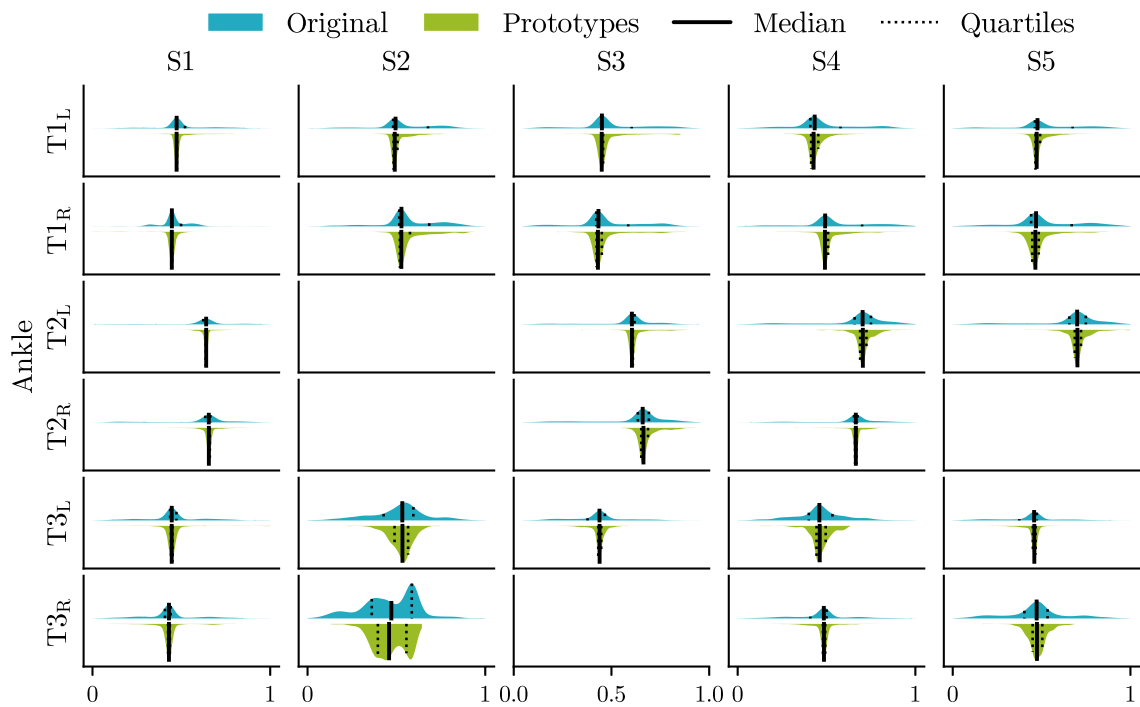


Figure 4.6: Mean (*black solid line*), 1st and 3rd quartiles (*black dotted lines*), and distributions of prototype (*green*) and original torques (*cyan*) of the ankle joint.

4.3 SELECTED CHANNELS

The selection of channels is assessed qualitatively with topographic maps of the distribution of EEG electrodes in the scalp of subjects according to the 10-20 system. The said maps for each subject and movement are shown in figures 4.7, 4.8, and 4.9 for hip, knee and ankle joints, respectively. Each topographic map shows the normalized occurrence of each channel given by the solutions of all executions of the GA for each subject and movement.

The summary statistics of the percentage of extracted prototypes and the number of selected channels for each movement and limb are shown in Table 4.1.

Table 4.1: Summary statistics of the percentage of extracted prototypes and number of selected channels.

Measure	Prototypes in %				Number of channels			
Statistic	Avg.	Min	Max	Std.	Avg.	Min	Max	Std.
Hip								
<i>Movement T1</i>								
Left	45.20	38.80	58.90	0.04	7	3	10	1.47
Right	42.20	38.50	52.80	0.02	8	3	12	2.46
<i>Movement T2</i>								
Left	40.60	27.60	55.80	0.10	6	3	9	1.54
Right	41.60	34.50	46.80	0.04	6	3	10	1.97
<i>Movement T3</i>								
Left	47.60	36.00	54.90	0.06	6	1	12	2.61
Right	49.60	38.60	59.70	0.06	7	3	11	2.11
Knee								
<i>Movement T1</i>								
Left	52.90	46.50	63.10	0.05	7	4	12	1.88
Right	47.30	43.90	54.90	0.02	7	2	11	2.46
<i>Movement T2</i>								
Left	52.50	46.10	60.30	0.04	7	3	14	2.11
Right	51.90	39.20	58.90	0.07	6	1	10	2.22
<i>Movement T3</i>								
Left	48.30	34.50	56.10	0.07	7	3	12	2.08
Right	50.40	39.30	58.20	0.05	7	2	13	2.60
Ankle								
<i>Movement T1</i>								
Left	48.80	41.60	60.90	0.06	7	3	13	2.13
Right	45.40	41.80	51.50	0.02	7	3	14	2.15
<i>Movement T2</i>								
Left	60.10	52.80	65.70	0.04	9	4	13	2.31
Right	57.20	54.10	59.80	0.01	6	2	8	1.39
<i>Movement T3</i>								
Left	48.00	41.90	54.10	0.04	6	3	12	1.74
Right	46.80	31.10	54.50	0.08	7	3	12	2.22
Avg	48.69	40.38	57.05	0.05	7	3	12	2.08

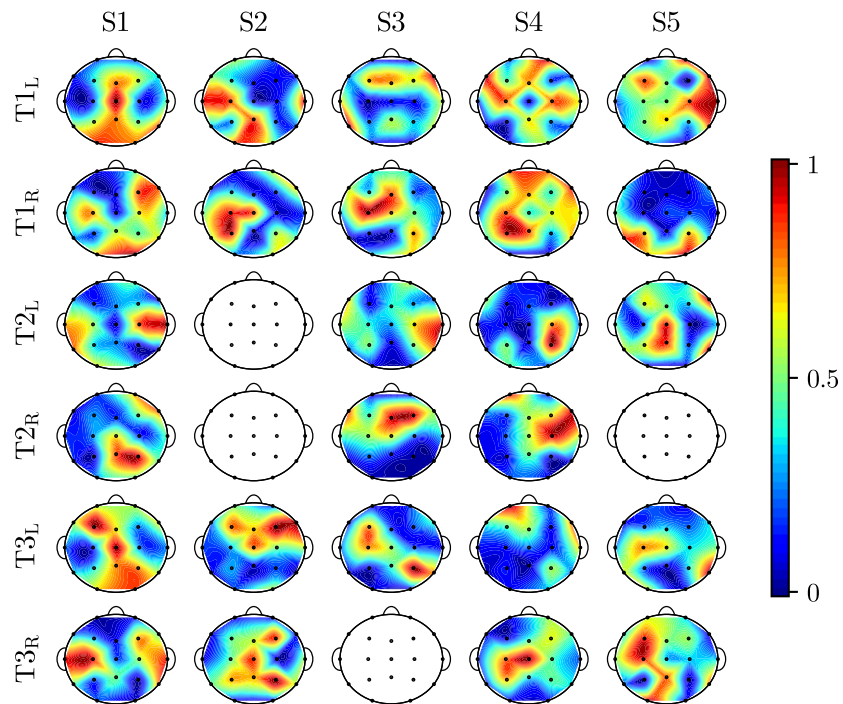


Figure 4.7: Normalized occurrence of EEG channels (*black dots*) selected along the executions of the GA for the hip joint.

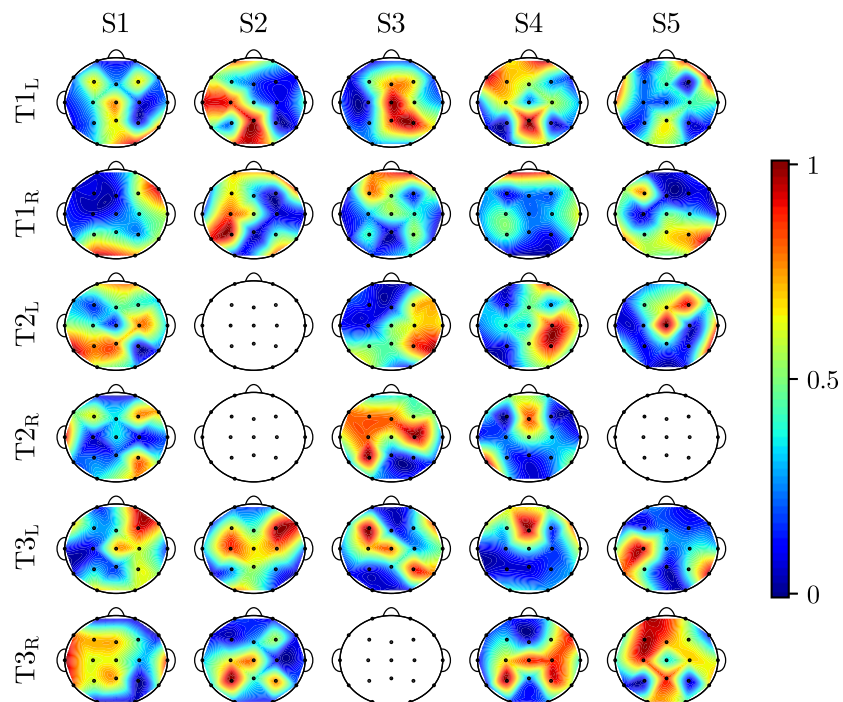


Figure 4.8: Normalized occurrence of EEG channels (*black dots*) selected along the executions of the GA for the knee joint.

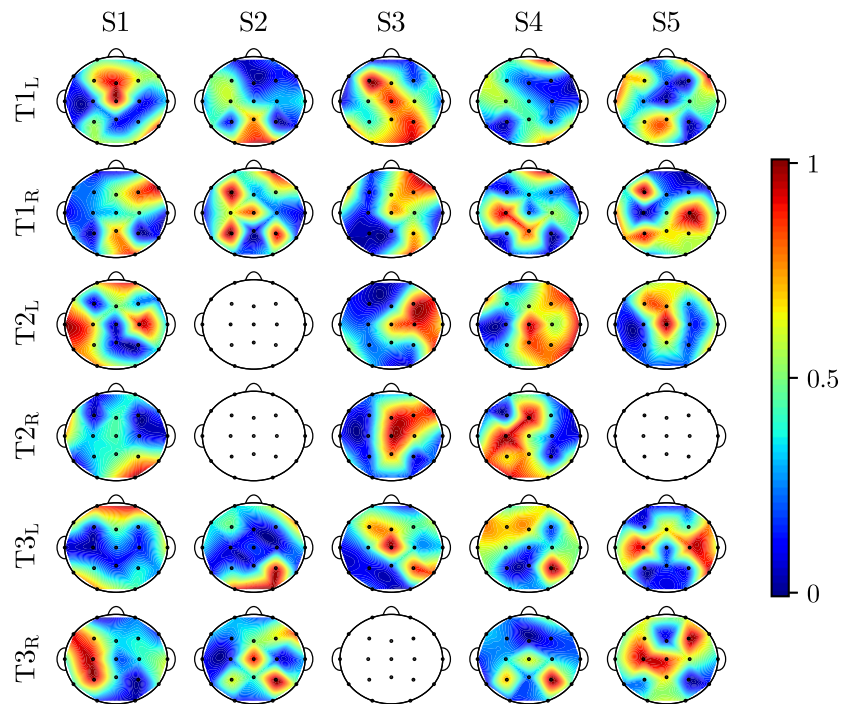


Figure 4.9: Normalized occurrence of EEG channels (*black dots*) selected along the executions of the GA for the ankle joint.

4.4 ESTIMATED TORQUES

Curves of estimations with highest R^2 values of optimized prototype torques, result of the optimized EEG channels used as inputs to the MLP, are shown in figures 4.10, 4.11, and 4.12. Figure 4.13 shows some estimation curves of prototypes overlapped with the entire curve of torques chosen for testing the MLP. The goodness of fit of the regression models trained with prototype torques were evaluated with R^2 and SNR measures, these measures are depicted in Figure 4.14 with respect to joints, movements and limb, and in Figure 4.15 with respect to subjects.

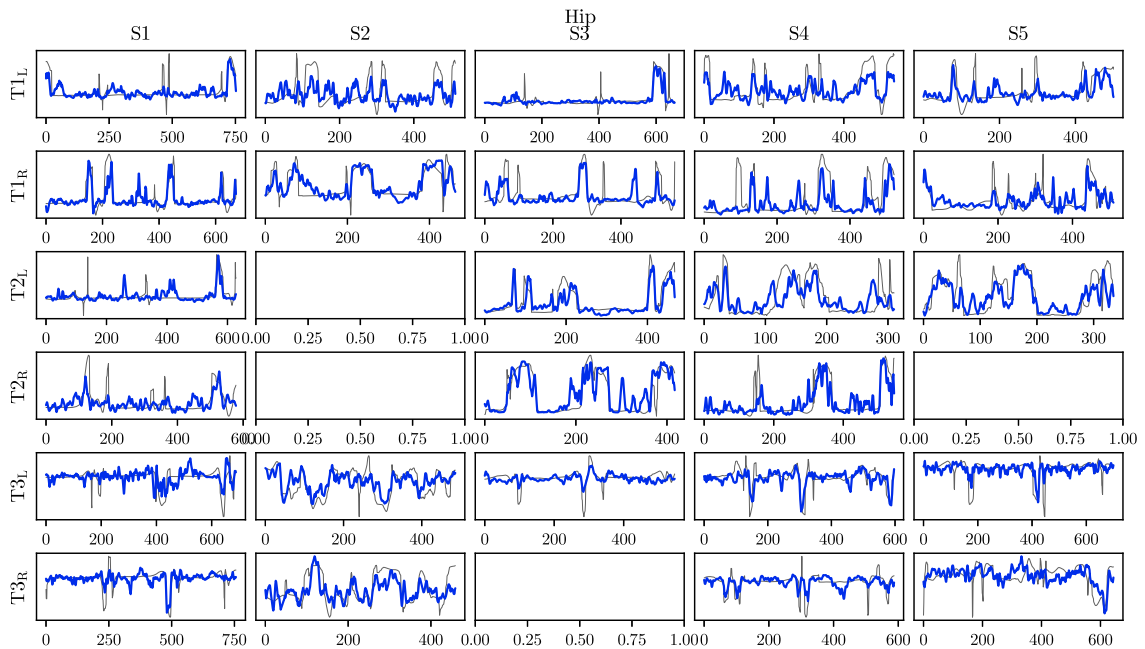


Figure 4.10: Estimation curves (*gray*) with the highest R^2 values of prototype hip torques (*black*).

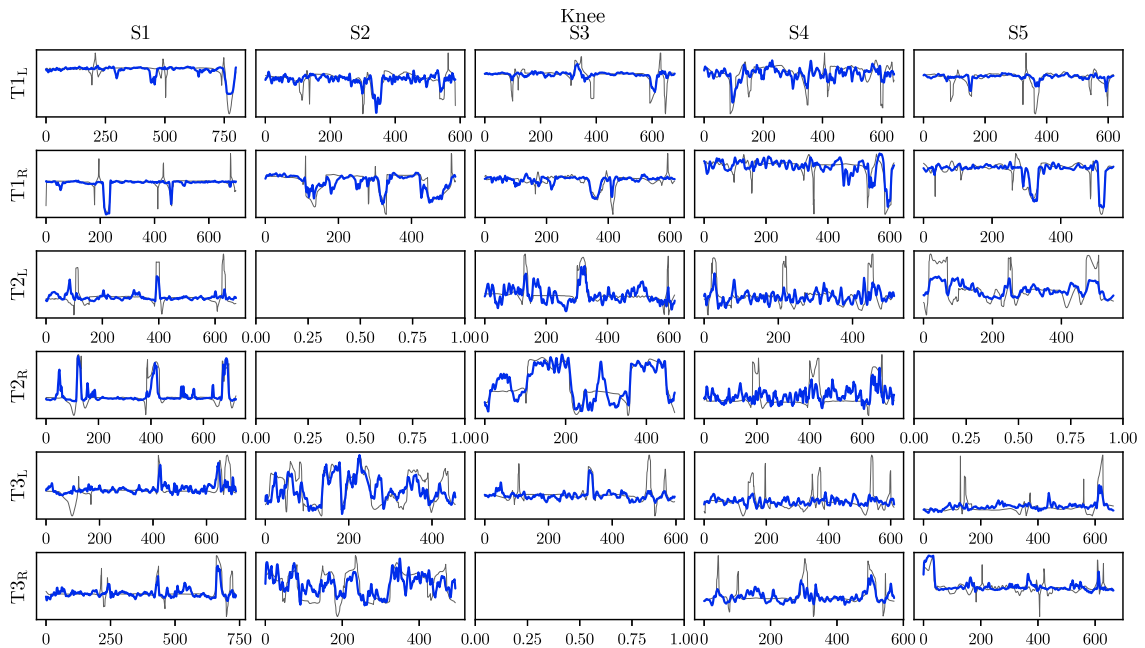


Figure 4.11: Estimation curves (*blue*) with the highest R^2 values of prototype knee torques (*gray*).

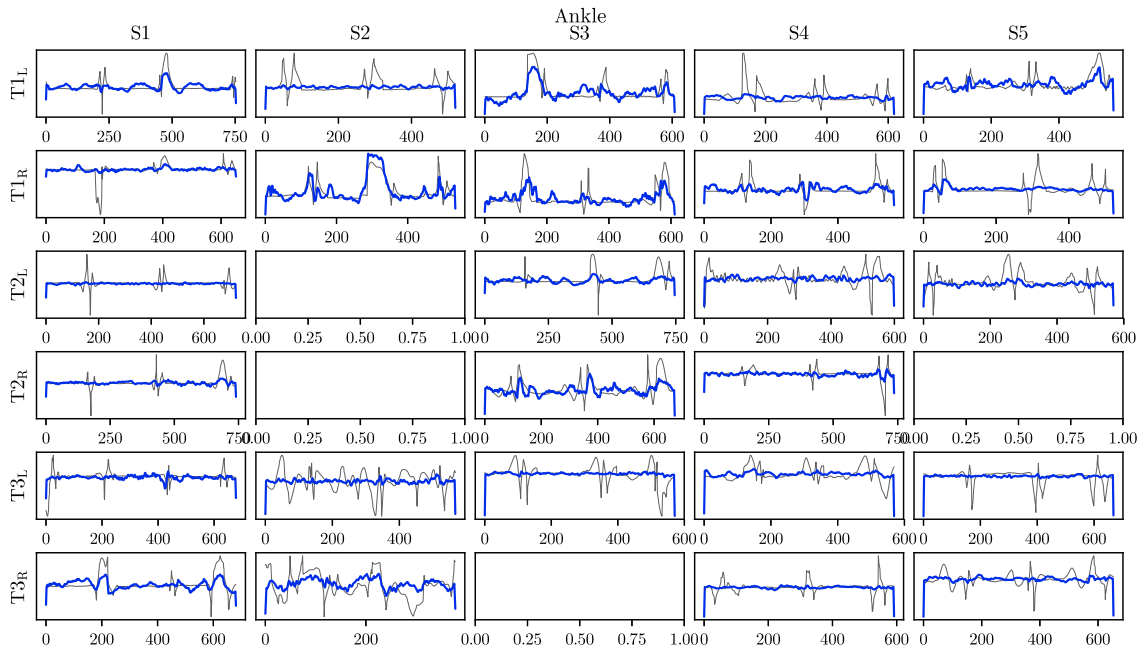


Figure 4.12: Estimation curves (*blue*) with the highest R^2 values of prototype ankle torques (*gray*).

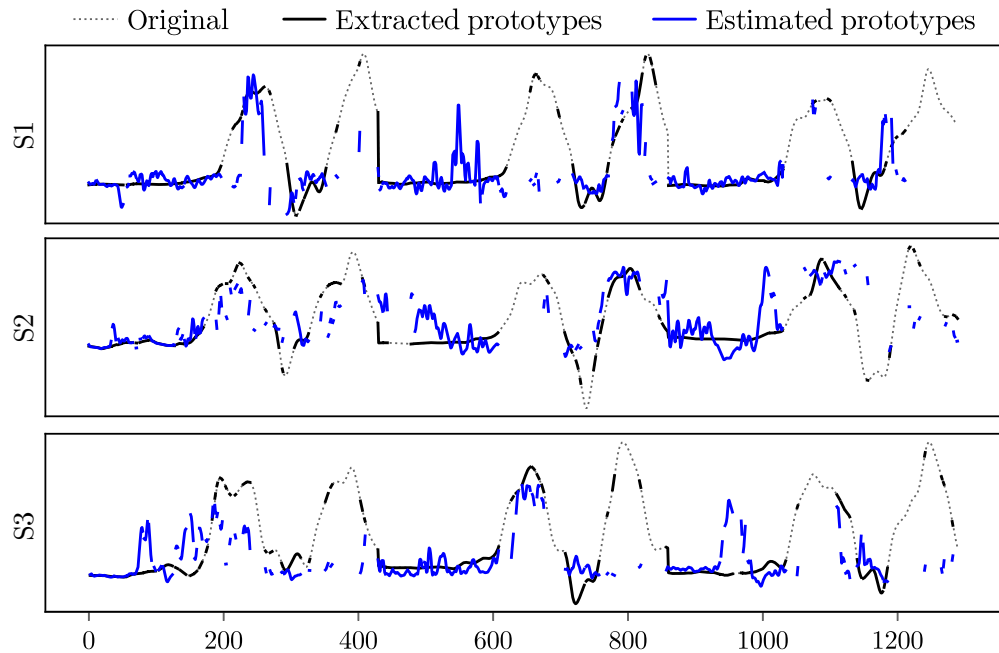


Figure 4.13: Estimation curves (*blue solid line*) of prototype torques (*black solid line*) of subjects 1, 2 and 3 overlapped to the entire curve of hip torques of movement $T1_R$ (*black dotted line*).

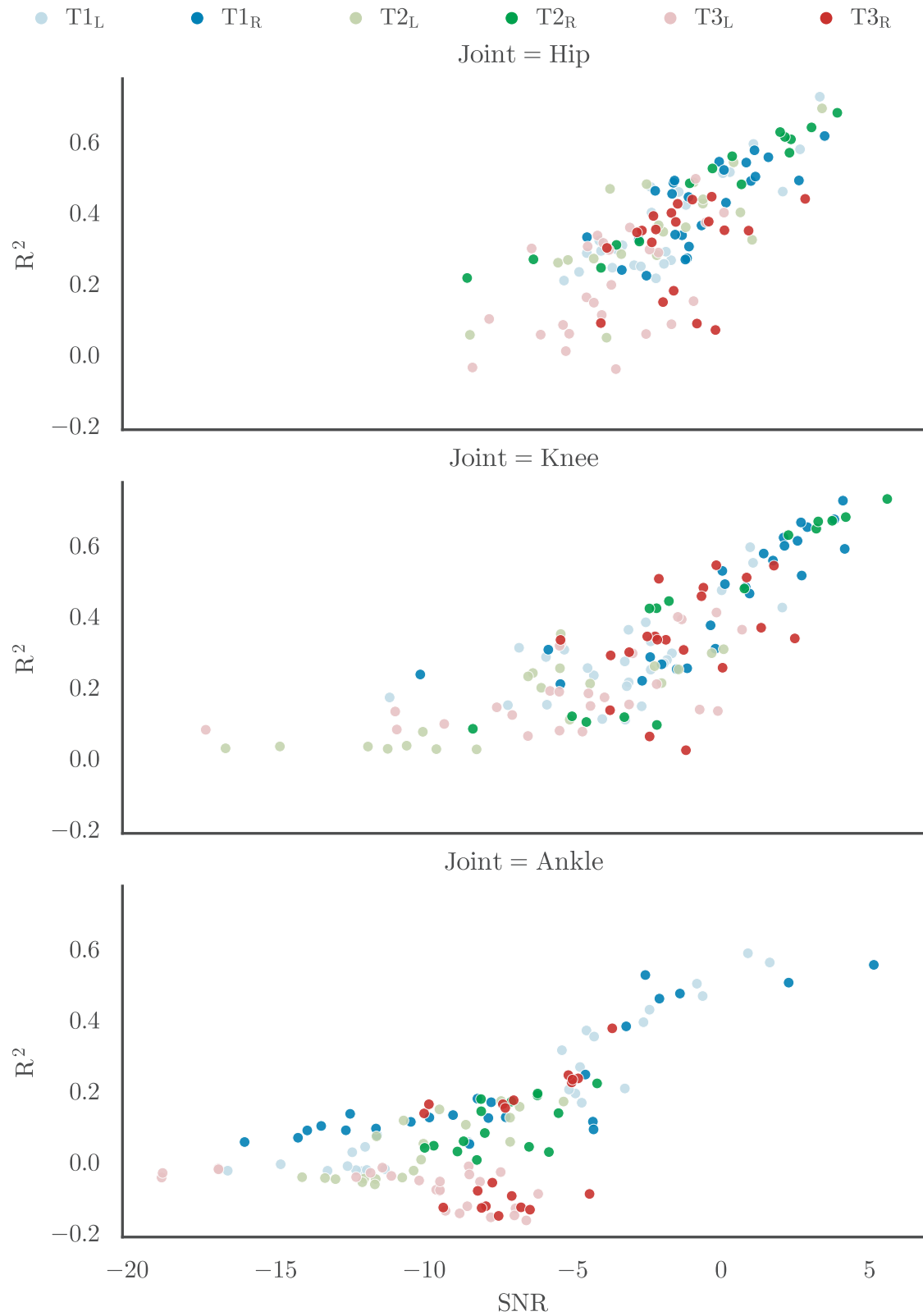


Figure 4.14: R^2 and SNR statistics of the top 5 estimations, according to R^2 , of prototype torques with respect to joint (*pairs of axes*), type of movement (*blue, green or red dots*) and left or right limb (*light and dark dots*).

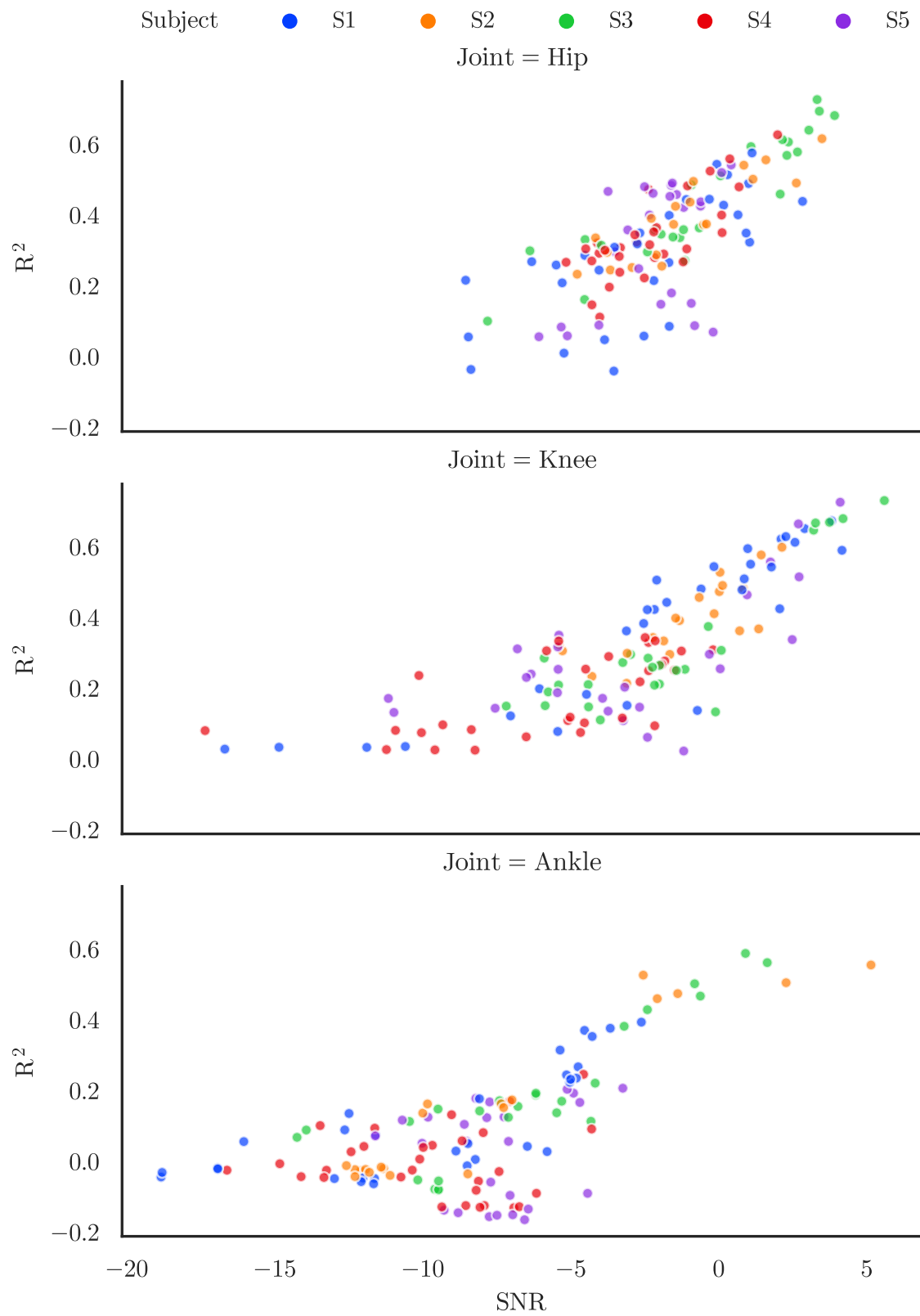


Figure 4.15: R^2 and SNR statistics of the top 5 estimations, according to R^2 , of prototype torques with respect to joints (*pairs of axes*) and subjects (*dot color*).

DISCUSSION

5.1 ON THE EXTRACTION OF PROTOTYPES

An initial objective of the research was to propose a scheme for identifying prototypes that characterize the dataset. The results from section 4.2 showed that the amount of prototypes extracted was about 50% of the data. A possible explanation for this might be the presence of noise and undetectable artifacts for the artifact-removal method used among the acquired EEG data. These artifacts could be present in the EEG due to input impedances and unshielded equipment, which is explained in [23, 24], as well as to the noisy nature of the brain signal at resting state, suggested in [25]. Alternatively, it may be the case that the data that could be extracted represents the time-invariant information of a brain dynamics strongly suspected to be time-variant, as [26–28] have argued. Whatever the case may be, the extraction of prototypes with C-KNN is, generally, consistent among several executions of the algorithm, which suggest the existence of subsets of EEG data formed by patterns that may represent the time-invariant information of the EEG with respect to the torque applied in the lower limbs during the execution of gait initiation movements.

Visually, there was noticeable difference in the quality of prototypes extracted from the torques of the ankle joint compared to the quality of the hip and knee joints, according to the distribution of the datasets, where prototype torques of the ankle joint seem to be concentrated around the mean. These significant differences indicate that most of the extracted ankle data was that of the basal phase of the protocol of movement, which means that information at the tails of the distribution was not satisfactorily extracted for the ankle joint. This result may be explained

by the fact that no indication was given to the subjects as to whether they should perform any defined movement in the knee joint, hence the lack of a defined pattern.

5.2 ON THE SELECTION OF CHANNELS

The second objective sought to propose optimal subsets of EEG channels for representing the process and for using later in regression tasks. The results given by the GA in section 4.3 suggest that, in average, only about 5 to 9 of the 19 available channels of EEG recordings were sufficient to represent the dataset.

Upon visual inspection of figures 4.7 to 4.9, the most recurrent channels given by the GA seem to be located in the frontal, pre-motor and motor regions of the brain for the torques of hip and knee joints. Such results are consistent for example with those reported by Zhang et al. [29] who, in the aim of decoding user's intended gait states, found that the most prevalent regions during the decoding were the frontal and fronto-central ones. These findings also match with those of neurosciences, in which the role of the motor cortex in the execution and planning of lower limb movements has been hypothesized for a long time, based on several studies dating back to the 1950s. However, the activation regions result from the optimization of ankle joint torques varies significantly among subjects, movements and limbs, so no claim can be made about the selections of channels for this joint, as expected according to the results of the previous section, which shows a poor prototype extraction for this joint.

5.3 ON THE ESTIMATION OF TORQUE

The estimation curves depicted in figures 4.10 and 4.11, corresponding to the hip and knee joints, show generally successful estimations of prototype torques that have been optimized with the proposed scheme. In contrast to this, however, the estimation curves in Figure 4.12 show poor tracking of the optimized ankle joints. These differences in the estimation of torques are easily more quantitatively seen from Figure 4.14, where the points corresponding to the estimation of ankle joint torques show lower measures of goodness of fit than those of the knee joint and even lower measures than those of the hip joint. This supports the claims made earlier in

this chapter about the hypothesized origin of the poor results found for the ankle torques: subjects did not pay special attention to the movement made in this joint compared to the attention paid to the hip and knee joints. On the other hand, an unexpected result was that tasks $T1$ and $T2$, which correspond to the step forward and step back movements, seem to have the best measures of goodness of fit in comparison to task $T3$, which corresponds to the step up movements; although it is consistent with the fact that the hip joint obtained the best results, probably due to the prevalence of the movement of this joint in these tasks.

Another interesting point to be addressed about the findings reported in the aforementioned depiction of goodness of fit is the significantly better estimations of right lower limb torques compared to the estimations of left lower limb torques. The laterality tendencies of the subjects to the right-footedness may explain this result, since the estimation is made in prototypes extracted from repetitions of movements made with the lower limbs, and the right-footedness might cause clumsiness in the execution of movements with the left lower limb, thus affecting the estimation of torques in these limbs.

For instance, whilst still not conclusive, the outperforming measures of goodness of fit of subjects S2 and S3 may suggest that an important factor in obtaining useful data for this type of analysis (EEG-movement) may be the attention paid by each individual in the execution of trials for this kind of experiments.

Additionally, from Figure 4.13, it could be seen that the estimation of extracted prototypes of torques represents points in the dataset that are part of the various phases of the movement, such as the increase, decrease and maintenance of the magnitude of torque.

CHAPTER 6

CONCLUSIONS

In summary, lower limb joint torques were successfully estimated from EEG signals, recorded from gait initiation movements, through the proposed scheme, which consisted of training a MLP neural network with prototypes extracted optimally with a GA under the assumption that these extracted data represent the relationship between the EEG and the torque.

Overall, this study suggests that patterns of EEG and torque signals can be successfully extracted, although not with a outstanding amount of data but with that necessary to make good estimations of relevant points of the output signal. The analysis of the characteristics of the extracted data will require further study, but has opened new pathways of understanding the functional relationship between acquired EEG and torque signals, even more, between neural and kinetic activity.

The research also supports the generalized idea of defining concisely the attention that must be given by the subjects in the protocol that they will be following, as a form to ensure the providing of useful data for further analysis.

It must be borne in mind that this study was only conducted on data of a small group of subjects over relatively short periods of time for each execution of the optimization algorithm for each selection of EEG channels. Further research is hence needed, to determine the specific set of channels necessary for the estimation of torque, on a larger number of subjects with more robust optimization algorithms before generalized conclusions can be drawn. However, this preliminary results support the path of using EEG signals as variables to be decoded for estimating user's intentions to perform motor tasks.

BIBLIOGRAPHY

- [1] Centers for Disease Control and Prevention (CDC), “Disability and functioning (noninstitutionalized adults aged 18 and over),” 2016. Accessed May 2019.
- [2] Department of Social Development (SEDESOL), “Diagnosis on the situation of people with disabilities in mexico,” 2016. Accessed May 2019.
- [3] N. A. Fitzsimmons, “Extracting kinematic parameters for monkey bipedal walking from cortical neuronal ensemble activity,” *Frontiers in Integrative Neuroscience*, vol. 3, 2009.
- [4] A. Presacco, R. Goodman, L. Forrester, and J. L. Contreras-Vidal, “Neural decoding of treadmill walking from noninvasive electroencephalographic signals,” *Journal of Neurophysiology*, vol. 106, pp. 1875–1887, oct 2011.
- [5] J. M. Antelis, L. Montesano, A. Ramos-Murguialday, N. Birbaumer, and J. Minguez, “On the usage of linear regression models to reconstruct limb kinematics from low frequency EEG signals,” *PLoS ONE*, vol. 8, p. e61976, apr 2013.
- [6] D. Liu, W. Chen, Z. Pei, and J. Wang, “Detection of lower-limb movement intention from eeg signals,” in *2017 12th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pp. 84–89, IEEE, 2017.
- [7] A. Kilicarslan, S. Prasad, R. G. Grossman, and J. L. Contreras-Vidal, “High accuracy decoding of user intentions using eeg to control a lower-body exoskeleton,” in *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 5606–5609, IEEE, 2013.

-
- [8] J. T. Gwin and D. P. Ferris, “An EEG-based study of discrete isometric and isotonic human lower limb muscle contractions,” *Journal of NeuroEngineering and Rehabilitation*, vol. 9, no. 1, p. 35, 2012.
- [9] M. Seeber, J. Wagner, R. Scherer, T. Solis-Escalante, and G. Mller-Putz, “Reconstructing gait cycle patterns from non-invasive recorded low gamma modulations,” in *Proceedings of the 6 th International Brain-Computer Interface Conference 2014*, 2014.
- [10] W. A. MacKay, “Wheels of motion: oscillatory potentials in the motor cortex,” *Motor cortex in voluntary movements: a distributed system for distributed functions*, vol. 181, p. 211, 2005.
- [11] T. Stöckel, R. Jacksteit, M. Behrens, R. Skripitz, R. Bader, and A. Mau-Moeller, “The mental representation of the human gait in young and older adults,” *Frontiers in Psychology*, vol. 6, jul 2015.
- [12] J. Perry, J. R. Davids, *et al.*, “Gait analysis: normal and pathological function,” *Journal of Pediatric Orthopaedics*, vol. 12, no. 6, p. 815, 1992.
- [13] H. H. Jasper, “The ten-twenty electrode system of the international federation,” *Electroencephalogr. Clin. Neurophysiol.*, vol. 10, pp. 370–375, 1958.
- [14] D. L. Schomer and F. L. Da Silva, *Niedermeyer’s electroencephalography: basic principles, clinical applications, and related fields*. Lippincott Williams & Wilkins, 2012.
- [15] H. Shibasaki and M. Hallett, “What is the Bereitschaftspotential?,” *Clinical Neurophysiology*, vol. 117, pp. 2341–2356, nov 2006.
- [16] T.-P. Jung, S. Makeig, C. Humphries, T.-W. Lee, M. J. McKeown, V. Iragui, and T. J. Sejnowski, “Removing electroencephalographic artifacts by blind source separation,” *Psychophysiology*, vol. 37, pp. 163–178, mar 2000.
- [17] H. SAKOE and S. CHIBA, “Dynamic programming algorithm optimization for spoken word recognition,” pp. 159–165, 1990.

- [18] T. Zhang, R. Ramakrishnan, and M. Livny, “BIRCH,” in *Proceedings of the 1996 international conference on Management of data*, ACM Press, 1996.
- [19] P. J. Rousseeuw, “Silhouettes: A graphical aid to the interpretation and validation of cluster analysis,” *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, nov 1987.
- [20] T. Kumar, “Solution of linear and non linear regression problem by k nearest neighbour approach: By using three sigma rule,” in *2015 IEEE International Conference on Computational Intelligence & Communication Technology*, IEEE, feb 2015.
- [21] K. Hechenbichler and K. Schliep, “Weighted k-nearest-neighbor techniques and ordinal classification,” *discussion paper*, vol. 399, 01 2004.
- [22] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *International Conference on Learning Representations*, 12 2014.
- [23] T. C. Ferree, P. Luu, G. S. Russell, and D. M. Tucker, “Scalp electrode impedance, infection risk, and EEG data quality,” *Clinical Neurophysiology*, vol. 112, pp. 536–544, mar 2001.
- [24] T. Ball, M. Kern, I. Mutschler, A. Aertsen, and A. Schulze-Bonhage, “Signal quality of simultaneously recorded invasive and non-invasive EEG,” *NeuroImage*, vol. 46, pp. 708–716, jul 2009.
- [25] M. E. Raichle, “The restless brain,” *Brain connectivity*, vol. 1, no. 1, pp. 3–12, 2011.
- [26] L. Leistritz, K. Schiecke, L. Astolfi, and H. Witte, “Time-variant modeling of brain processes,” *Proceedings of the IEEE*, vol. 104, pp. 262–281, feb 2016.
- [27] S. Druckmann and D. B. Chklovskii, “Neuronal circuits underlying persistent representations despite time varying activity,” *Current Biology*, vol. 22, pp. 2095–2103, nov 2012.

-
- [28] S. Druckmann and D. B. Chklovskii, “Neuronal circuits underlying persistent representations despite time varying activity,” *Current Biology*, vol. 22, no. 22, pp. 2095–2103, 2012.
- [29] Y. Zhang, S. Prasad, A. Kilicarslan, and J. L. Contreras-Vidal, “Multiple kernel based region importance learning for neural classification of gait states from EEG signals,” *Frontiers in Neuroscience*, vol. 11, apr 2017.