Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE

Corso di Laurea Magistrale in Matematica

PATTERN RECOGNITION METHODS FOR EMG PROSTHETIC CONTROL

Tesi di Laurea in Analisi Matematica

Relatore:
Chiar.ma Prof.ssa
GIOVANNA CITTI

Correlatori:
Prof.
DAVIDE BARBIERI
Ing.
EMANUELE GRUPPIONI

Presentata da:
SIMONA BACCHERINI

II Sessione
Anno Accademico 2015/2016

"Io stimo più il trovar un vero,
benché di cosa leggiera,
che 'l disputar lungamente delle massime questioni
senza conseguir verità nissuna."
Galileo Galilei

Introduction

Scope of this thesis is to study instruments of pattern recognition with application to EMG prosthetic control. The majority of prostheses of upper limb used nowadays are active myoelectric prostheses.
The fundamental object in myoelectric control is EMG (electromiography) signal, which represents the "electrical manifestation of neuromuscolar activation associated with a contracting muscle" [14]. This kind of signal is mainly measured by non invasive surface electrodes and it activates the device when it passes over a fixed threshold, allowing the prosthesis movement by an electric motor placed in it.
The basic steps in a pattern recognition method are the measurements of the object of interest and the definition of features able to describe them, feature extraction and classification. It is common to have to deal with large dataset of measurements, described by multiple variables mutually correlated. Thus, when features are defined, it is usually applied a dimensionality reduction algorithm such as Principal Component Analysis [11]. At the end, in supervised learning, a classification is provided, which in the particular case of EMG problems predicts the intended movements performed by the device. Other efficient classification algorithms are Support Vector Machine [4] [1] and Neural Networks, which seem to be the best choice to classify EMG data [16].
After giving a detailed review of the mentioned methods and others, we propose a new classification method based on two main assumptions: firstly, the electrodes location plays a role in pattern discrimination. Secondly, we suggest an approach different from the one proposed in [16], basically based on the analysis of each patient without any previous training phase on able-bodied subjects. It comes as a result of comparison in the Fourier analysis between able-bodied and trans-radial amputee subjects. The different approach of considering five time domain features together with each electrode contribute separately has revealed an average classification accuracy equals to 75%. We consider this result a promising

starting point for the computation of a classification algorithm for the control of EMG prosthetic device.
This study was done in collaboration between the Math Department of University of Bologna, the Math Department of University Autonoma of Madrid and Research and Training Area of Centro Protesi INAIL in Vigorso di Budrio (BO).
The thesis is organized as follows:

- in Chapter 1 we review the nature of electromiography signals, the mathematical model which describes them and the way in which these signals can be detected. We analyze the state-of-the-art in prosthetic control, describing the most commercially prosthetic hands based on myolectric approach. We then report the fundamental steps in a pattern recognition based system control, focusing in particular on the time domain features used in [16];

- in Chapter 2 we firstly introduce the Principal Component Analysis dimensionality reduction method, based on the orthogonal transformation of the axes which consists in the resolution of an eigenvalues/vectors problem. We then describe three main classification algorithm, widely used in prosthetic control and for each one we review the main results present in literature. In particular, we describe Support Vector Machine, the hyperplane classification method which from linearly separable data can be extended to non separable data, Clustering methods with particular focus on the k-means algorithm and the recent field of spectral clustering, which are deep analyzed in [23]. Finally, in the last section we describe Artificial Neural Networks, starting from the perceptron or logistic regression which has one input and one output neurons and extend it to multi-layer perceptron, which is characterized by hidden neurons and for which we describe the training algorithm of back-propagation;

- in Chapter 3 we report the results of our research. Firstly, we describe the preliminarly good results in the Fourier space based on three able-bodied subjects, from the point of view of the FFT signal trend decay and as the result of a frequency based classification algorithm composed by PCA dimensionality reduction and k-means in the reduced subspace spanned by the first three principal components. Since the same approach fails when applied to trans-radial amputee subjects, we propose a different classification algorithm which considers each electrodes separately, computing for

each one five time domain features. We report our classification results and make considerations on them. Finally, we propose some future developments based on time windowing, data shuffle partition for the validation of the model and an automatic features selection procedure as a minimization problem.

Introduzione

Scopo di questa tesi è studiare strumenti di pattern recognition con applicazione al controllo protesico EMG. Al giorno d'oggi, la maggir parte di protesi per arto superiore in uso sono protesi attive mioelettriche.
L'oggetto fondamentale nel controllo mioelettrico è il segnale EMG (elettromiografico), che rappresenta la manifestazione elettrica di un'attivazione neuromuscolare associata ad una contrazione muscolare [14]. Tale tipologia di segnale viene tipicamente misurata per mezzo di elettrodi superficiali non invasivi e consente l'attivazione della protesi quando il segnale supera una soglia fissata, consentendo il movimento del dispositivo per mezzo di un motore elettrico in esso posizionato.
I passi fondamentali in un metodo di pattern recognition sono le misurazioni dell'oggetto di interesse e la definizione di features capaci di descrivere tali oggetti, l'estrazione di features e la classificazione. Tipicamente si lavora con dataset di misure dimensionalmente grandi, descritte da molteplici variabili mutualmente correlate. Pertanto, definite le features, si applicano algoritmi di riduzione della dimensionalità come la Principal Component Analysis [11]. Infine, nell'apprendimento supervisionato, si fornisce una classificazione, che nel caso particolare di problemi EMG predice i movimenti pianificati eseguiti dalla protesi. Altri algoritmi efficienti di classificazione sono Support Vector Machine [4] [1] e Reti Neurali, le quali sembrano essere la scelta migliore per classificare dati EMG [16].
Dopo avere fornito una dettagliata revisione dei metodi citati ed altri, proponiamo un nuovo metodo di classificazione basato su due ipotesi principali: in primo luogo, il posizionamento degli elettrodi riveste un ruolo per il riconoscimento di patttern. In secondo luogo, suggeriamo un approccio differente rispetto a quello proposto in [16], basato ossia sull'analisi singola di ciascun paziente senza alcuna fase preliminare di training su soggetti normodotati. Esso giunge come risultato dall'analisi di Fourier tra pazienti normodotati e soggetti amputati transradiali. Il diverso approccio di considerare cinque features temporali insieme

al contributo di ciascun sensore separatamente ha riportato una accuratezza di classificazione in media pari al 75%. Consideriamo tale risultato un promettente punto di partenza per l'implementazione di un algoritmo di classificazione per il controllo di una mano protesica mioelettrica.
Il seguente studio è stato svolto in collaborazione tra il Dipartimento di Matematica dell'Università di Bologna, il Dipartimento di Matematica dell'Università Autonoma di Madrid e l'Area Ricerca e Formazione del Centro Protesi INAIL di Vigorso di Budrio (BO).
La tesi è organizzata come segue:

- nel Capitolo 1 si analizza la natura dei segnali elettromiografici, in particolare il modello matematico che li descrive e il modo in cui tali segnali vengono rilevati. Analizziamo lo stato dell'arte riguardo al controllo protesico, descrivendo le mani protesiche basate sul controllo mioelettrico maggiormente in commercio. In seguito, riportiamo i passaggi fondamentali che caratterizzano un sistema di controllo basato sulla pattern recognition, rivolgendo particolare attenzione alle features temporali utilizzate in [16];

- nel Capitolo 2 introduciamo in primo luogo il metodo di riduzione della dimensionalità Principal Component Analysis, basato su una trasformazione ortogonale degli assi che consiste nella risoluzione di un problema agli autovalori/autovettori. Descriviamo poi tre tra i principali algoritmi di classificazione, ampiamente utilizzati nel controllo protesico e per i quali riportiamo alcuni risultati centrali presenti in letteratura. In particolare, descriviamo Support Vector Machine, un metodo di classificazione a iperpiani che a partire dalla versione lineare inerente dati separabili linearmente può essere esteso alla classificazione di dati non separabili, metodi di clustering con particolare attenzione all'algoritmo k-means e al recente settore di clustering spettrale, analizzato nel dettaglio in [23]. Infine, nell'ultima sezione vengono descritte le Reti Neurali Artificiali, assumendo come punto di partenza il percettrone o logistic regression costituito da un neurone di input ed uno di output, ed estendendolo al multi-layer perceptron, caratterizzato da neuroni nascosti per i quali descriviamo l'algoritmo di apprendimento back-propagation;

- nel Capitolo 3 riportiamo i risultati ottenuti in questa ricerca. In primo luogo, descriviamo i risultati positivi ottenuti nello spazio di Fourier relativi

a tre soggetti normodotati, dal punto di vista del decadimento della FFT e come risultato di un algoritmo di classificazione nel dominio delle frequenze, composto dall'algoritmo di riduzione di dimensionalità PCA e k-means nel sottospazio ridotto generato dalle prime tre componenti principali. Poichè tale approccio fallisce se applicato a soggetti amputati transradiali, proponiamo un diverso algoritmo di classificazione che considera separatamente ciascun elettrodo, calcolando per ciascuno di essi cinque features temporali. Riportiamo i risultati di classificazione ed elenchiamo alcune considerazioni su di essi. In conclusione, proponiamo possibili sviluppi futuri basati su di una suddivisione in finestre temporali, partizione per mescolamento dei dati per la validazione del modello ed una procedura di selezione automatica di features come un problema di minimizzazione.

Contents

List of Figures

List of Tables

Chapter 1

EMG pattern recognition for prosthetic control

In this chapter we introduce the fundamental motivations of the thesis, that are electromiography (EMG) signal and its use for prosthetic control.
We first describe in Section 1.1 EMG signal from the anatomical and physiological point of view, in order to have a general comprehension of the nature of the signal and of the way it is produced. Then we describe the basic analysis techniques and the mathematical model. Section 1.2 and 1.3 contain a review of the state of the art in prosthetic control by pattern recognition, a field in great development in the last decades. The last Section 1.4 presents some specifications on the data used during this study, resulting from a stage at INAIL Centro Protesi in Vigorso di Budrio (BO).

1.1 Electromiography signal

The nervous system is the motor of human body, consisting in elementary cells called neurons that rapidly communicate with different parts of the body by electric signals. Every voluntary and involuntary action is produced by the nervous system.

A muscle is a soft tissue compound of cells, which can change in shape and length and it can generate and transmit force. The coordinated activation of muscles provides posture and produces both voluntary and involuntary movements. There are three types of muscle tissue in human body: skeletal muscle, smooth muscle and cardiac muscle. Our interest is focused on skeletal muscle, whose tissue is attached to the bone and its contraction is responsible for supporting

and moving the skeleton [14].
The fundamental functional unit of a muscle is the motor unit and it can generate a motor unit action potential (MUAP) when it is activated from the nervous system. Whenever the action or the force is required, the activation of a motor unit becomes continuous and this produces the motor unit action potential trains (MUAPT).
In [5] an EMG signal is defined as the "electrical manifestation of the neuromuscolar activation associated with a contracting muscle", while in [14] it is described as "the train of motor unit action potential (MUAPT) showing the muscle response to neural stimulation".

- The MUAPT may be expressed as

$$u_i(t) = \sum_{k=1}^{n} h_i(t - t_k) \tag{1.1}$$

where $h(t)$ is a filter that represents the shape of the MUAP. Furthermore,

$$t_k = \sum_{l=1}^{k} x_l, \quad for \quad k, l = 1, ..., n$$

represents the time locations of the MUAPs, x represents the interpulse intervals (time between adjacent MUAPs), n is the total number of interpulse intervals in a MUAPT.

- The EMG signal may be expressed by a linear summing of MUAPTs, as shown in the following expression

$$m(t, F) = \sum_{i=1}^{p} u_i(t, F) \tag{1.2}$$

where $u_i(t)$ represents the MUAPT defined in Eq. (1.1) and F is the force generated by the muscle.

1.1.1 How to detect myolectric signals

Electromiography is defined as the discipline that studies the muscle electrical signals as the result of muscles contraction [14]. In particular, surface electromiography is a non-invasive technique for measuring the electrical activity of

skeletal muscles. In this section we describe different types of electrodes, the principal devices used for detecting EMG signal and some typical precautions in signal processing.

The most common way for revealing and evaluating electrical muscle activity is via surface electromiography using electrodes. There are two main types of electrodes: surface electrodes and inserted electrodes.
The first are widely used because of their non-invasive nature. Depending on the construction, surface electrodes can be divided into active or passive electrodes. Active electrodes contain a high input impedance electronics, while passive electrodes consist of conductive detection surface that reveals the current on the skin via its interface [5].
Nowadays, active electrodes are preferred because of their light mass and small size; therefore they detect the electrical activity of a group of MUAP and not of the single ones.
The inserted electrodes can be wire or needle electrodes. The most common is the needle electrode, where one or more needles are inserted under the skin and the cannula containing the needle remains inserted in the muscle for the time of the test.
The reason why needle electrodes are preferred is the small size that enables the devices to detect individual MUAP, another reason is the possibility of repositioning them within the muscle. However, wire electrodes are less painful than needle ones. On one hand, it is not necessary to maintain them into the muscles fibers throughout the duration of the test, on the other they tend to move from the original insertion at the beginning of contractions.
Obviously, the choice of the type of electrode depends on the particular application. In our case, we consider only surface active electrodes (more detailes can be found in Section 1.4).

1.2 State of the art in prosthetic hand

This work concerns the control of an active prosthetic hand via EMG signals, that is nowadays the most common way of control an artificial device. In this section we present an overview on the causes of amputation and we illustrate different level of amputation. Our interest is aimed at upper limb amputation, which is largely common in Europe and US. We then present the developments occured in prosthetic, in particular on myolectric devices.

1.2.1 Amputation

Amputation is the removal of part of a limb by surgery, illness or trauma. It is frequent to distinguish between different level of amputation, depending on the characteristics of the remaining stump. According to [17], on average every year there are more than 18,000 upper limb amputations considering Europe and US, the most frequent of them are trans-humeral (28%), digit (22%), trans-radial (19%) and partial hand amputations (19%).
This type of amputations are mainly due to traumatic, malignant or vascular causes. Upper limb amputations are not as frequent as lower limb. Because of the extreme complexitiy of human hand, the reliability of a prosthetic hand is a topic in continuous development and different solutions of these devices are offered to patients.

1.2.2 Prosthetic hands and myolectric control

A prosthesis is an artificial device that replaces a part of the body missing.
It is possible to classify upper limb prostheses on the base of prosthesis functionality, that means in which way the prosthesis is activated. We can make the classification as shown in Table (1.1).
Most commercially available upper limb prostheses are body-powered or electrical motor powered. Body-powered prostheses enjoy benefits as low cost and high reliability, whenever they are operated by movements of the amputees' body through cables and sometimes manual control. Therefore, this type of devices are limited in utility and are slow to operate, although they are widely used.
We focus on myolectric prostheses, that are a competent alternative for mechanical body-powered systems [13].
Myolectric control approach was firstly proposed in 1940s but the first commercialized myolectric hand was developed in 1960. The control strategy of upper-limb myolectric prostheses uses EMG signals to control the prosthetic devices. The EMG signals are measured with electrodes, usually located on a pair of agonist/antagonist muscles, that reveals the remnant muscle activity. When the EMG amplitude is greater than a certain threshold, the associated prosthetic movement is performed by an electric motor placed in the device.

Passive prosthesis	**Active prosthesis**
Widely commercially available, characterized by the pinch gesture	Recover the functionality of missing limb
Cosmetic Reconstruction of missing limb aesthetic, mainly made up with silicone	*Body powered (cinematic)* Device powered by remnants mucles that activate cables for moving prosthesis components
Working Pinch or hook that allows people to work, without aesthetic aspects	*Extra powered* More advanced, translate electrical energy in mechanical energy for moving prosthesis components
	Myoelectric control based on EMG signal
	Electric control activated by a switch

Table 1.1: Functional classification of upper limb prosthesis

The use of myolectric prostheses have several advantages over body-powered prostheses. For instance, it does not use invasive techniques for signal detecting and the muscle activity required for prosthetic control is relatively small. Other important aspects of controllability in myolectric control are the accuracy of movement selection, the intuitiveness of actuating control and system response time [13]. However, only the 50% of patients use that kind of devices [8].
An example of myolectric hand can be seen in Figure (1.1):

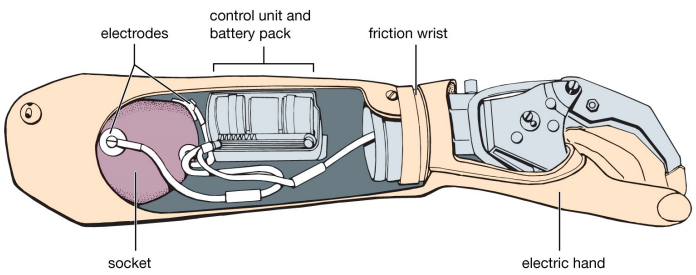

Figure 1.1: Upper limb myolectric prosthesis

1.2.3 Polyarticular hands

Prothesic hand tries to be a compromise between functionality, aesthetic and affordability. The main goal is to be as close as possible to human hand which has 21 freedom degrees and it is able to make precision and force movements.
Different models of such devices are commercially available, but we focus on the principal classes of *tridigits hand* and *polyarticular hand*. In the first, the thumb and the index and set of medium are placed in opposition, while these ones dragged passively ring finger and little finger. Therefore, tridigit hand allows only one type of grasp, without any sensory feedback [22]. This device is severley limited compared to human hand.
However, in the last decades polyarticular hand prototypes have been proposed, with the aim of solving problems of control, sensory feedback and dexterity. From 2007, these laboratory hands has been transformed in devices available for patients and usable in everyday life, and pattern recognition techniques has been used to control them (as described in Section 1.3).

Figure 1.2: Example of myolectric and tri-digit prothesic hand

We review the principal polyarticular hands commercially available (see Figure (1.3)):

- iLimb: it is the first polyarticular hand, characterized by five fingers individually active connected to an aluminum frame that simulates the backbone of the palm. This hand can realize 8 principal grasps, with closure velocity proportional to EMG signal amplitude;
- Bebionic: polyarticular hand that allows 14 movements with quick closure velocity;

- Vincent: it is the first touch sensing hand prosthesis, which allows an active individual agility of the fingers and the thumb. It is characterized by the lateral movement of the opposable thumb to the ring finger, making it versatile and interesting in terms of the development of new control strategies.

Figure 1.3: Example of polyarticular hands: iLimb, Bebionic, Vincent (from left to right)

1.3 Pattern recognition in prosthetic control

A significant innovation in prosthetic control is the use of pattern recognition techniques based on EMG signals. This new approach is due to use pattern classification techniques for detecting pattern and extract hidden signal informations. Using a pattern classification technique, different pattern can be obtained from EMG signals and used to identify the intended movements. Therefore, once a pattern has been detected, the prosthesis is activated and the desired movement is performed with the highest possible rate of accuracy. In general, an EMG pattern recognition-based prosthetic control approach consists of *EMG measurements, feature extraction and classification.*

At first, EMG measurements are performed in order to capture more and reliable myolectric signals, then features are extracted from data to retain the most important discriminating information from the EMG signals, and at the end a classification algorithm is applied to predict the intended movements [8].

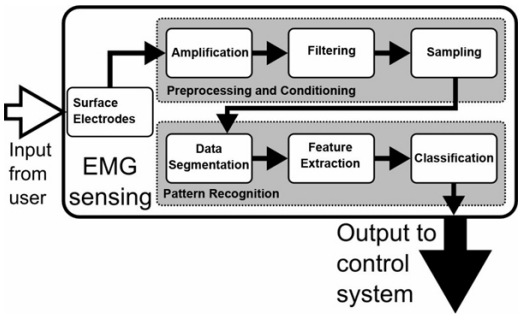

Figure 1.4: Pattern recognition-based system control

1.3.1 EMG measurements

The reason why pattern recognition is used for upper limb prostheses control is that the forearm contains the residual wrist muscle and some residual hand muscles, allowing therefore wrist and hand control movements.
The use of EMG signal as an estimator of motor intent is deeply affected by electrodes location and electrodes number. While in a healthy subject the mucles position is known, every amputee presents specific characteristics on muscle structure. Moreover, it is known that a progress in upper-limb prosthetic has been the *target muscle reinnervation*, a surgical procedure introduced by Dr. Todd Kuiken consisting in the selective transfer of brachial nerves to new muscle sites [18]. The consequence is that the new EMG sites defined in this way contain a mixture of functions corresponding to different nerves.
As reported in [5], the location of the electrodes should be done according to signal/noise ratio, signal stability and cross-talk from adjacent mucles. With reference to signal/noise ratio, when electrodes are placed on the skin they reveal a signal composed by all the action potential of the muscles under the device. The resulting signal is commonly very noisy and difficult to manage. This is due to motion artefacts, the electrode equipment noise and the floating ground noise. Therefore, to obtain more useful information, it is required that electrodes remove as much as possible the noise content. In this work, we consider pre-amplified electrodes as Ottobock 13E200 sensors.
For myolectric transradial prostheses, the EMG signals are usually measured with a number between 8 and 18 electrodes, typically placed around the circumference of the stump [8]. Many studies demonstrate that the use of a high number of electrodes increase the EMG patter recognition performance, but in [1] a study

based on 4 sensors correctly placed and a light signl processing can give high classification accuracy comparable with systems with more sensors. Overall, the preferred location of a surface electrode is in the halfway between the center of the innervation zone and the further tendom [5].
Moreover, EMG pattern recognition is performed on windowed data: from each window a classification decision will be made. The window length is usually 100-250 ms, mainly with overlapped windows. More generally, window length is chosen according to patient's skill and real time application [8].

1.3.2 Feature extraction

The recording of an EMG signal is presented as a time sequence. It is not practical to pass it directly to the classifier, also beacuse of high dimensionality of data recorded. Therefore, the sequence must be mapped into a smaller dimension vector, called *feature vector*. In other words, an EMG pattern associated to a limb movements is described with the feature vector extracted from EMG recordings.
Furthermore, the success of any pattern recognition classification problem depends almost on the selection or extraction of features.
There are three main feature categories: time domain, frequency (spectral) domain, and time-scale (time-frequency) domain.

Because of their computational simplicity and intuitiveness, *time domain features* are the most popular in myolectric control and are mainly based on signal amplitude [13].
In [16], after data segmentation using the overlapped windowing technique, the following features are extracted from each time window:

- Mean (M): it represents the average value of the EMG amplitude

$$M = \frac{1}{W}\sum_{i=1}^{W} x_i \tag{1.3}$$

- Root Mean Square (RMS): it represents the mean power of the signal

$$RMS = \sqrt{\frac{1}{W}\sum_{i=1}^{W} x_i^2} \tag{1.4}$$

- Willison Amplitude: it represents the number of counts for each change in

the EMG signal amplitude that exceeds a predefined threshold

$$WA = \frac{1}{W}\sum_{i=1}^{W-1} f(|x_i - x_{i-1}|),\ f(x) = \begin{cases} 1, x \geq threshold \\ 0,\ otherwise \end{cases} \tag{1.5}$$

- Slope sign change (SSC): it represents the number of times the slope of the EMG signal changes spin

$$SSC = \frac{1}{W}\sum_{i=2}^{W-1} f[(x_i - x_{i-1}) \times (x_i - x_{i+1})],\ f(x) = \begin{cases} 1, x \geq threshold \\ 0,\ otherwise \end{cases} \tag{1.6}$$

- Simple square integral (SSI): it represents the area under the curve of the squared signal

$$SSI = \sum_{i=1}^{W} x_i^2 \tag{1.7}$$

- Variance (V): it is a statistical measure thar represents how signal varies from its average value M

$$V = \frac{1}{W-1}\sum_{i=1}^{W}(x_i - M)^2 \tag{1.8}$$

- Waveform length (WL): it represents the cumulative length of the EMG signal waveform

$$WL = \sum_{i=1}^{W-1} |x_{i+1} - x_i| \tag{1.9}$$

Spectral or frequency analysis is mostly used to study muscle fatigue and needs more computational resources in comparison to time-domain features. Power spectral density plays a fundamental role in spectral analysis; it is defined as a Fourier transform of the autocorrelation function of a signal. Its two characteristic variables, the mean and median frequency, give some information about signal spectrum and its change over time.

Time-frequency analysis can be used for signal de-noising, identifying fatigue in long-term activity and isolating coordinated muscle activities, [13]. In this approach, the aim is to maintain both time and frequency content. Since Fourier transform loses signal time domain information and so the time-localization of a

specific event, it can be used the Short-time Fourier transform. This takes into account time and frequency, mapping a signal into a two-dimensional function of them. Another common tool is Wavelet analysis, that reveals data aspects as trends, breakdown points, discontinuities in higher derivatives and self-similarity [13].

1.3.3 Classification

In this last stage, linear or non linear algorthms (*classifier*) assign the extracted features to the class they most probably belong to. In Chapter 2 we give a detailed review about the most important results in literature on classification methods on EMG signals for myolectric control.

1.4 Acquisition of EMG signals

In this section we finally describe the experimental setup, giving detailes about materials and dataset obtained from measurements. The results obtained from these data are explained in Chapter 3.

We consider both able-bodied subjects and amputee subjects. Three healthy subjects, aged between 24 and 28, and twenty trans-radial amputee, aged between 18 and 65, free of known musclar and/pr neurological diseases, partecipated in the experiments. For healthy subjects, data were recorded from the dominant arm, while for amputees it was asked which hand they preferred in daily activities. Six commercial active sEMG sensors (Ottobock 13E200=50) were placed on the subjects' forearm using a silicone bracelet. Sensors were equidistantly placed in the bracelet: for healthy subjects, it was located about 5 cm below the elbow, while for amputees it was placed on the circumference of the stump, about 5 cm below the elbow. Sensors operate in 4-8.5 V, bandwidth of 90-450 Hz. Data were collected using an acquisition system and transmitted to the PC via USB.

The subjects were sitting in front of a monitor interface, showing $np = 5$ different gestures: rest (hand relaxed), fist (hand with all fingers closed), pinch (hand with thumb and finger touching), spread (hand open), pointing (hand with all fingers closed with only index pointing), as depicted in Fig. (1.6).

Each gesture was randomly repetead $nc = 10$ times, with a recording time T equals to 2s. Since the sEMG signals are sampled at Fs frequency, from each gesture registration we obtain a $L \times ns$ matrix, where $L = Fs \cdot T$ data and $ns = 6$ number of sensors.

The overall matrix referred to the acquisition of the sequence of hand gestures

has dimension $np \times L \times ns$, where $np = 5$ gestures. Finally, as the gestures are repeated nc times, we obtain a dataset matrix $nc \times np \times L \times ns$ for each subject. An example of EMG signal recorded from surface electrodes on amputee subject is shown in Fig. (1.5).

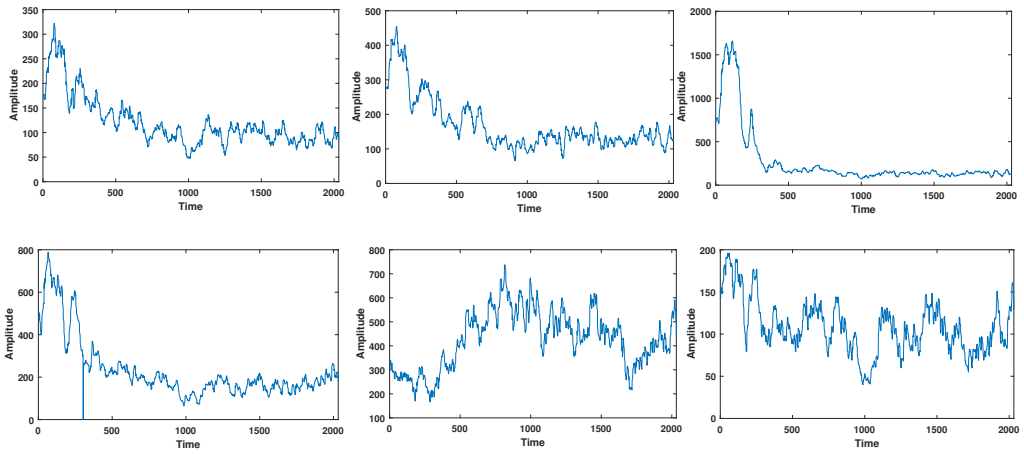

Figure 1.5: EMG signal from amputee subject: from left to right signals recorded from sensors 1-6

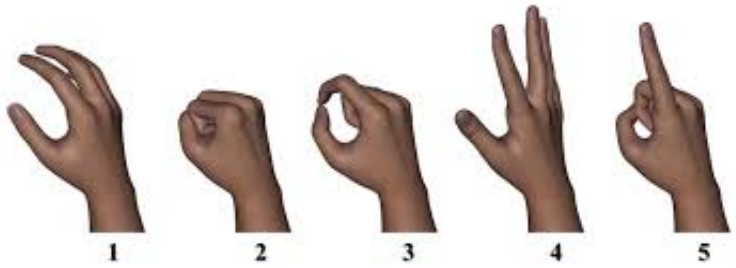

Figure 1.6: The five hand gestures: rest (1), fist (2), pinch (3), spread (4), pointing (5)

Chapter 2

Classification methods

In this chapter we introduce the most common classification methods used in pattern recognition based system control, with reference to EMG prosthetic control. In Section 2.1 we describe the Principal component analysis (PCA) algorithm, that is an unsupervised dimensionality reduction method widely used that does not make use of class information. Section 2.2 contains the explaination of the Support vector machine (SVM), a linear classification algorithm, while in Section 2.3 we review the clustering methods with focus on the K-means clustering. Section 2.4 is about a different clustering method, based on graph theory and related to probabilistic notions, that is Spectral clustering. In conclusion, in Section 2.5 we describe Artificial Neural Network with details on the back-propagation algorithm.

2.1 Principal component analysis

The first stage in a pattern recognition system is dimensionality reduction. It is known that this technique is necessary to extract important informations for class discrimination.

There exist two main dimensionality reduction categories, according to the functional cost to be minimized: *feature selection* methods and *feature extraction* methods. On the one hand, feature selection methods try to determine the best subset of the original features set, using a metric to evaluate the features subset. The simplest way is to find the best subset that minimizes the error rate. On the other hand, feature extraction methods attempt to determine a new set of variables as a linear combination of the original features that best represent the original ones. A common method belonging to this category is Principal

component analysis (PCA). These methods can be summarized as in the below scheme:

$$\textit{feature selection}: \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \rightarrow \begin{pmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{in} \end{pmatrix}, \quad \textit{feature extraction}: \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \rightarrow \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix} = f \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

Principal component analysis (PCA) is the oldest statistical technique of dimensionality reduction. It was first introduced by Pearson in 1901 and has several applications in statistical data analysis, pattern recognition and data compression.

It is a multivariate statistics algorithm which implements a data orthogonal transformation. Its aim is to reduce the dimensionality of multivariate measurements finding a smaller set of uncorrelated variables able to give a good representation of the data. In order to reach this goal, it performs a coordinate rotation such that the new axes are the ones with maximum variance, projecting the data onto the eigenvectors of the covariance matrix. The new variables so obtained are called Principal Components (PCs). Because PCA does not make use of class information, it may be regarded as an unsupervised method but it does not guarantee that the axes with maximum variance contain good features for classification.

2.1.1 PCA algorithm

We consider a random vector x with n elements. Typically, the elements of x are measurements mutually correlated and therefore there is some redundancy in x.

As preliminary step, the vector x is centered subtracting its mean $E[x]$, that means

$$x \longleftarrow x - E[x]$$

Then, x is linearly transformed into a different vector y of m components, with $m < n$, such that redundancy is removed. This is done finding an orthogonal transformation such that the elements of x in the new coordinates become uncorrelated. An important aspect in applications is that computing y from x does not require high computational resources, beacuse of the linearity of the PCA algorithm that we are going to describe.

Definition 2.1 (PCA problem)**.**

For every vector x of components $x_1, ..., x_n$, and for every vector w_1 of scalar

weights $w_{11}, ..., w_{n1}$, we set

$$y_1 = w_1^T x = \sum_{j=1}^{n} w_{j1} x_j \tag{2.1}$$

We define y_1 as the first principal component if its variance is maximally large. The solution of the PCA problem is a vector w which maximizes the variance of y_1.

To establish the weight vector w_1, using the properties of the expected value $E[\cdot]$ and the definition of the covariance matrix C, we have to maximize the following functional cost

$$J(w_1) = E[y_1^2] = E[(w_1^T x)^2] = w_1^T E[xx^T] w_1 = w_1^T C w_1, \ s.t. \ ||w_1|| = 1 \tag{2.2}$$

The constraint $||w_1|| = 1$ is due to fact that the variance of y_1 depends on the norm and orientation of w_1 and it grows as the norm grows.
Assume $w_1 = (w_{11}, ..., w_{1n})$, its norm $||w_1||$ is defined as the Euclidean norm

$$||w_1|| = (w_1^T w_1)^{\frac{1}{2}} = \left(\sum_{j=1}^{n} w_{j1}^2 \right)^{1/2}$$

The matrix C in Eq. (2.2) is the $n \times n$ covariance matrix of x, defined as

$$C = E[xx^T]$$

The solution of the PCA problem is given in terms of the unit-length eigenvectors $e_1, ..., e_n$ of the covariance matrix C. The eigenvectors are in descendent order, such that the corresponding eigenvalues $d_1, ..., d_n$ satisfy $d_1 \geq d_2 \geq ... \geq d_n$.
More precisely, the solution maximizing Eq.(2.2) is given by $w_1 = e_1$, therefore the first principal component of x is $y_1 = e_1^T x$.
If we generalize the criterion in Eq.(2.2) from one single principal component to m components, with $m < n$ and similarly denoting the m-th component $y_m = w_m^T x$, we have to generalize also the constraint of uncorrelation. This means that y_m must be uncorrelated with all the previous principal components.
From the preliminary assumption of zero means on x, we can observe that

$$E[y_m] = E[w_m^T x] = w_m^T E[x] = 0$$

Therefore, also the m-th principal component have zero means.
The conditions of zero means for both x and y_m implie that y_m is uncorrelated with all the previously found principal components if and only if

$$E[y_m y_k] = 0, \ k < m \tag{2.3}$$

Therefore,

$$E[y_m y_k] = E[(w_m^T x)(w_k^T x)] = w_m^T E[xx^T] w_k = w_m^T C w_k \overset{(2.3)}{=} 0, \ k < m$$

In particular, for $m = 2$, we have to maximize $E[y_2^2] = E[(w_2^T x)^2]$ in the subspace orthogonal to the first eigenvector e_1 of C. Thus, the previous equation is maximized by $w_2 = e_2$.
Recursively, we obtain $w_k = e_k \implies y_k = e_k^T x$ is the k-th principal component of x. In other words, the principal components are defined as weighted sums of elements of x with maximum variance, where the weights are the ordered eigenvectors of the covariance matrix of x. It also follows that the variance of the principal components are given directly by the eigenvalues $d_1, ..., d_n$ of C.

The question that arises immediately is how many principal components have to be considered. One approach can be the choice of the principal components that represents about the 80-90% of global variability. Another is based on the decreasing of the eigenvalues sequence. Hence a threshold is fixed and it determines how many principal components can be used (see Figure(2.1)).
Most of the time, a rather small number of principal components are sufficient.

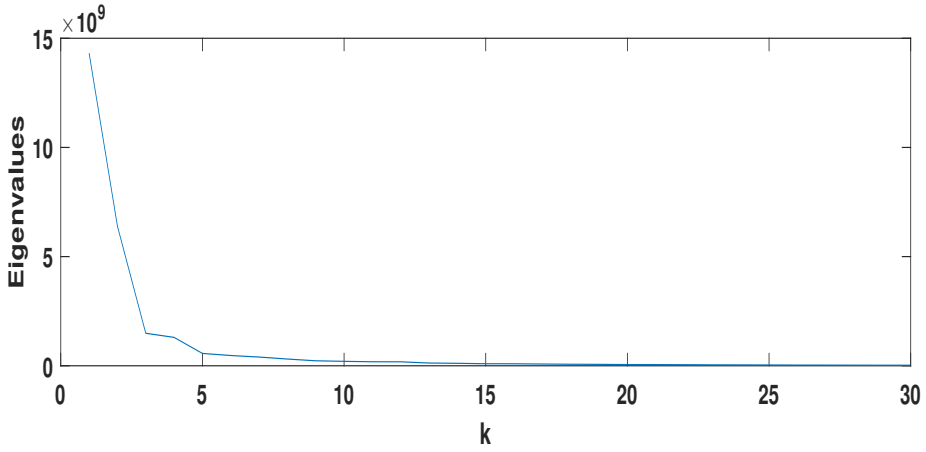

Figure 2.1: Example of PCs number selection. The screen plot shows that 5 PCs are enough

We have shown that the principal components are defined as weighted sums of the elements of the vector x with maximum variance, under the constraints that the weights are normalized and the principal components are uncorrelated with each other. This formulation is related to another way to pose the PCA problem, that is the *minimun mean-square error compression of* x. In these terms, we

search a set of m orthonormal basis vectors such that the mean-square error between x and its projection onto the subspace spanned by the m vectors is minimal.

In fact, if we assume that $w_1, ..., w_m$ are the orthonormal basis vectors, that means $w_i^T w_j = \delta_{ij}$, the projection of the vector x on the subspace with dimensionality equals to m is denoted by

$$x_\perp = \sum_{i=1}^{m} (w_i^T x) w_i$$

Therefore, the vectors $w_1, ..., w_m$ must minimize the mean-square error criterion defined by

$$\begin{aligned} J_{MSE} &= E\big[||x - x_\perp||^2\big] = E\big[||x - \sum_{i=1}^{m}(w_i^T x)w_i||^2\big] = \\ &= E[||x||^2] - 2E < x, \sum_{i=1}^{m}(w_i^T x)w_i > + E\big[||\sum_{i=1}^{m}(w_i^T x)w_i||^2\big] \\ &= E[||x||^2] - E\big[\sum_{i=1}^{m}(w_i^T x)^2\big] = tr(C) - \sum_{i=1}^{m} w_i^T C w_i. \end{aligned} \tag{2.4}$$

Then, the minimum of J_{MSE}, under the constraint of orthonormality of the w_i, is given by any orthonormal basis of the PCA subspace spanned by the m first eigenvectors $e_1, ..., e_m$. This solution does not specify the basis of the subspace, but states that any orthonormal basis of the subspace gives the same optimal compression.

It has been proved that if we change the orthonormality condition from (i) to (ii)

$$(i)\ w_i^T w_j = \delta_{ij} \quad \longrightarrow \quad (ii)\ w_j^T w_k = \lambda_k \delta_{jk},$$

with λ_k positive and different, then the mean-square error problem has a unique solution given by scaled eigenvectors.

2.1.2 On-line methods and recursive least-squares approach

We have seen that the fundamental object in computing the PCA is the resolution of the eigenvalues/vectors problem for the covariance matrix, which allows the definition of the principal components in terms of its eigenvectors. However, it is not always feasible to solve this problem with standard numerical methods, as QR method, mainly because of the computational resourses required when the dimensionalty n of C is large. Different apporaches have been proposed, in particular we mention the gradient ascent algorithms for solving the minimization problem in Eq. (2.2) and the PAST Algorithm.

Gradient ascent algorithms

One alternative to the resolution of eigenvalues/vectors problem for C may be the use of gradient ascent algorithms or on-line methods that minimize the PCA criterion in Eq. (2.2), finding the eigenvectors of the covariance matrix C. The main advantage is that these algorithms work on-line, therefore using each input vector one time, without computing the covariance matrix at all. This way represents the basis of the PCA neural network learning rules. The PCA network learns the principal components by unsupervised learning rules: it basically updates the weight vectors until they become orthonormal, making corrections on the eigenvectors. This kind of learning algorithm and the implementation in neural network is useful in feature detection and data compression problems. More detailes on these algorithms can be found in [11].

The PAST Algorithm

The Projection Approximation Subspace (PAST) Algorithm is a fast algorithm with low computational cost, largely used in signal processing and control. We review its implementation as an alternative resolution of the PCA problem. Recall the mean-square error criterion from Eq. (2.4). Having fixed a sample $\{x(j)\}_{j=1}^{T} \subset \mathbb{R}^n$, and using the matrix notation $W = (w_1, ..., w_m)^T \in \mathbb{R}^{n \times m},\ m \leq n$, J_{MSE} can be estimated as

$$\bar{J}_{MSE} = \frac{1}{T} \sum_{i=1}^{T} \left[||x(i) - W^T W x(i)||^2 \right]$$

The problem is now to find W recursively. The coefficient $\frac{1}{T}$ can be replaced by an exponential β^{t-i}, where $\beta \in [0, 1]$, obtaining, for $t \leq T$

$$J_{MSE}(t) = \sum_{i=1}^{t} \beta^{t-i} ||x(i) - W^T(t) W(t) x(i)||^2$$

Introducing $y(i) = W(i-1)x(i)$, we can approximate the previous expression and the modified least-squares criterion becomes

$$J'_{MSE} = \sum_{i=1}^{t} \beta^{t-i} ||x(i) - W^T(t) y(i)||^2$$

The PAST algorithm, proposed by Yang, can be summarized in the following steps [11]:

$$
\begin{aligned}
y(t) &= W(t-1)x(t) \\
h(t) &= P(t-1)y(t) \\
m(t) &= \frac{h(t)}{\beta + y^T(t)h(t)} \\
P(t) &= \frac{1}{\beta} Tri[P(t-1) - m(t)h^T(t)] \\
e(t) &= x(t) - W^T(t-1)y(t) \\
W(t) &= W(t-1) + m(t)e^T(t)
\end{aligned}
$$

where Tri stands for the upper triangular part of the current matrix. It is common to choose as initial values $W(0)$ and $P(0)$ the $n \times n$ unit matrices. The most complicated operation in this algorithm is division by a scalar, therefore no matrix inversion is required. It has also been proved that it has low computation cost and its convergence is rather fast. Because of this positive computational aspects, it is a competent alternative to the formal computation of the PCA problem via diagonalization of the covariance matrix.

2.1.3 PCA in EMG pattern recognition

In EMG pattern recognition systems, PCA is the main technique used in the pre-processing stage for dimensionality reduction. The reason why it is preferred is its simplicity in formulation and computation, as we have shown in the previous paragraphs, supported by a lot of good numerical results on experiments.
In particular, by projecting the data with the goal of determining new uncorrelated variables, the covariance structure is lost. Morover, if the data are dispersed in the original feature space, then the PCA tries to consolidate the informations more than feature selection does. Because PCA is an unsupervised method, there is no possibility to deteriorate the efficacy of dimensionality reduction with the knowledge of class membership [7].
An interesting analysis of PCA applied to myolectric signals is provided by [7], where Class separability (CS) criterion and Principal component analysis (PCA) are compared. The former uses the Euclidean distance for feature extraction and makes use of class membership information, while the latter does not make use of class information. Both techniques specify a subset of $k < n$ most informative features, where n is the dimensionality of the feature space. The study has shown

that PCA is widely superior to CS based dimensionality reduction, in particular when time-frequency representaion based features are used. Moreover, it has been proved that using a PCA/LDA (Linear discriminant analysis) combination, with the high-dimensional feature space of the Wavelet packet transform, have reached an average classification error of 6.25%. The positive result is that applying PCA does not degrade performance of an easy linear classifier as LDA rather than Multi layer perceptron (MLP), which has a more difficul implementation and training.
In [16], PCA and CSP (Common spatial pattern) are compared, revealing that the best/optimized feature vector for PCA preprocessing techinque is the pair Root Mean Square (RMS) - Willison Amplitude (WA) with an Artificial neural network (ANN) classifier, with the highest classification accuracy of 87.34±7.30%. Another application can be found in [4], where PCA is applied to five EMG signals measured by five surface electrodes revealing that they can be linearly reduced to two signals, losing on average $7.7 \pm 4.4\%$ of the signal variance. However, it must be observed that the reduction in the number of feature vectors does not mean that only two electrodes are sufficient to obtain the same results.

2.2 Support vector machine

Support vector machine (SVM) is a supervised machine learning algorithm proposed by V. Vapnik, both for pattern regression and classification, widely used since 1990s. It is basically a binary learning machine, that can be extended to more general problems. The main idea behind the method can be summed up as follows:
" Given a training set, the Support Vector Machine constructs a separating hyperplane between the two classes in order to have maximum margin [1] *."*
In this section we introduce the geometrical interpretation of SVM as a binary classification problem, then we will be able to extend it to the more general case of non-separable data. We define some central notions as support vectors and the inner-product kernels. At the end, we review some results occured in literature on the applications of that kind of classifier to EMG signal classification.

2.2.1 Geometrical interpretation of linearly separable data

Consider the training sample $D = \{(x_i, y_i)\}_{i=1}^{N}$, where $x_i \in \mathbb{R}^n$ represents the input pattern for the i-th example and $y_i \in Y$ is the corresponding desired

[1] The margin is defined as the maximal distance between the hyperplane and the set of data.

response. We denote Y the label set. In this context, we assume that the sample are linearly separable into two classes in the input space, with $Y = \{+1, -1\}$. Therefore, the equation of a separating hyperplane between the positive ($y_i = +1$) class and the negative ($y_i = -1$) one can be written as

$$w^T x + b = 0 \tag{2.5}$$

where $w \in \mathbb{R}^n$ is a weight vector and b is the bias.
Because of the assumption made on Y, we may write

$$w^T x_i + b \geq 0 \quad for \quad i \quad such \quad that \quad y_i = +1 \tag{2.6}$$

$$w^T x_i + b \leq 0 \quad for \quad i \quad such \quad that \quad y_i = -1 \tag{2.7}$$

For a given weight vector w and bias b, we denote with ρ the margin between the hyperplane and the data in the corresponding two regions. The goal of a support vector machine is to find a particular hyperplane in the form of Eq. (2.5) for which ρ is maximized. From this formulation, the decision surface is referred to be the optimal hyperplane. We refer to Fig.(2.2) for a graphical representation of the problem:

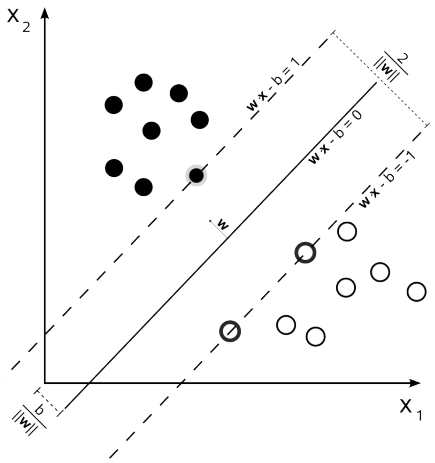

Figure 2.2: Linear SVM: optimal separating hyperplane between positive (black circular points) and negative class (white circular points).

Consider that (w_0, b_0) define the optimal hyperplane, it follows immediately that $w_0^T x + b_0 = 0$, for x on the hyperplane. We can denote $g(x) := w_0^T x + b_0$, as the linear discriminating function that gives an algebraic measure of the distance from x to the optimal hyperplane. If we consider an input vector x and the

optimal hyperplane, x can be decomposed as

$$x = x_\rho + r\frac{w_0}{||w_0||}$$

where x_ρ is the normal projection of x onto the hyperplane and r is the desired value of the algebraic distance. Therefore,

$$g(x) = g\left(x_\rho + r\frac{w_0}{||w_0||}\right) \overset{g(x_\rho)=0}{\Longrightarrow} g(x) = r||w_0|| \tag{2.8}$$

Primal problem

Now, given the training set $D = \{(x_i, y_i)\}_{i=1,\dots,N}$, (w_0, b_0) must satisfy also the following constraints

$$w_0^T x_i + b_0 \geq 1 \quad for \quad y_i = +1 \tag{2.9}$$

$$w_0^T x_i + b_0 \leq -1 \quad for \quad y_i = -1 \tag{2.10}$$

which can be obtained by rescaling the inequalities in Eq.(2.6)-(2.7), dealing with linearly separable data.
The data points (x_i, y_i) for which Eq. (2.9) or (2.10) are satistifed by the equality sign are called *support vectors*. They become the most important for the location of the optimal hyperplane: in fact, they are the closest vectors to the hyperplane and the most difficult to classify. Their number is usually much smaller than the total number of samples used in the training phase.
Consider, in conclusion, a support vector x^s. By definition, we have

$$g(x^s) = w_0^T x^s + b_0 = \mp 1, \quad for \quad y^s = \mp 1 \tag{2.11}$$

From Eq. (2.8), it follows that the desired algebraic distance is

$$r = \frac{g(x^s)}{||w_0||} = \begin{cases} \frac{1}{||w_0||}, & if \quad y^s = +1 \\ \frac{-1}{||w_0||}, & if \quad y^s = -1 \end{cases}$$

Then, denoting with ρ the optimal value for the margin of the hyperplane, it follows that

$$\rho = 2r = \frac{2}{||w_0||}$$

In summary, we have shown that the problem of maximizing the margin of the separating hyperplane can be reformulated as a minimization problem for the Euclidean norm of the weight vector w.

Combining the constraints in Eq.(2.9) with Eq.(2.10) in the compact form $y_i(w_0^T x_i + b_0) \geq 1$, the problem can be formulated as a constraint-optimization problem:
"Given the training set D, find the optimal values of (w,b), such that

$$y_i(w^T x_i + b) \geq 1, \quad with \quad w \quad minimizing \quad \phi(w) = \frac{1}{2} w^T w"$$

This formulation is called the *primal problem*. Therefore, the primal problem deals with a convex cost function $\phi(w)$ and linear constraints.
Introducing the method of Lagrangian multipliers, we can define the Lagrangian function

$$L(w, b, \alpha) = \frac{1}{2} w^T w - \sum_{i=1}^{N} \alpha_i \big[y_i(w^T x_i + b) - 1 \big] \tag{2.12}$$

where α_i are the Lagrangian multipliers, the first term in the r.h.s of Eq. (2.12) is the convex cost function $\phi(w)$ and the second term is the algebraic equations of the linear constraints. The solution of the constrained-optimization problem is determined by the saddle point of the function L. Thus, differentiating $L(w, b, \alpha)$ with respect to w and b and setting the results equal to 0, we obtain the following optimality conditions:

$$\begin{cases} \frac{\partial L(w,b,\alpha)}{\partial w} = 0 \\ \frac{\partial L(w,b,\alpha)}{\partial b} = 0 \end{cases}$$

Applying these conditions to the Eq. (2.12), we obtain

$$w = \sum_{i=1}^{N} \alpha_i y_i x_i, \qquad \sum_{i=1}^{N} \alpha_i y_i = 0 \tag{2.13}$$

In addition, for all the constraints that are not satisfied as equalities, the corresponding Lagrangian multiplier α_i must be zero. This means that only the multipliers α_i that exactely satisfy

$$\alpha_i \big[y_i(w^T x_i + b) - 1 \big] = 0$$

may assume non zero values, thus corresponding to the support vectors.

Dual problem

We can formulate another problem called the *dual problem*, which has the same optimal values of the primal one but for which the Lagrangian multipliers α_i provide the optimal solution.

Let us expand Eq.(2.12)

$$L(w,b,\alpha) = \frac{1}{2}w^T w - \sum_{i=1}^{N} \alpha_i y_i w^T x_i - b\sum_{i=1}^{N} \alpha_i y_i + \sum_{i=1}^{N} \alpha_i$$

Because of Eq.(2.13), we have

$$w^T w = \sum_{i=1}^{N} \alpha_i y_i w^T x_i = \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x_i^T x_j$$

Denoting $Q(\alpha) := L(w,b,\alpha)$, it follows that

$$Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x_i^T x_j \tag{2.14}$$

It can be noted that the functional $Q(\alpha)$ to be maximized depends only on the data input x_i, in particular on the dot products $x_i^T x_j$.
We can now state the dual problem as follows:
"Given the training set D, find the Lagrangian multipliers α_i that maximize the objective function Q(α) defined in Eq. (2.14), under the constraints

$$(1) \quad \sum_{i=1}^{N} \alpha_i y_i = 0; \qquad (2) \quad \alpha_i \geq 0, \quad \forall i = 1, ..., N"$$

It must be observed that the constraint (2) is satisfied by inequality sign for all the support vectors which have a non zero coefficient α_i, while it is satisfied with the equality sign for all the training data for which the α_i are zero. Therefore, having obtained the optimum value for the Lagrangian multipliers, denoted $\alpha_{0,i}$, we can find the optimum value for the weight vector w and for the bias b.
In particular, from Eq.(2.13) we have

$$w_0 = \sum_{i=1}^{N_s} \alpha_{0,i} y_i x_i \tag{2.15}$$

where N_s denotes the number of support vectors; from Eq.(2.11) it follows that

$$w_0^T x^s + b_0 = +1 \quad for \quad y_i = +1 \Longrightarrow b_0 = 1 - \sum_{i=1}^{N_s} \alpha_{0,i} y_i x_i^T x^s \tag{2.16}$$

To sum up, the separating hyperplane for separable data is defined by (w_0, b_0) just obtained in Eq. (2.15) and Eq. (2.16).
Finally, the previous argument is based on the following

Theorem 2.2.1 (Duality theorem, Bertsekas-1995)**.**
(i) If the primal problem has an optimal solution, the dual problem also has an optimal solution, and the corresponding optimal values are equal.
(ii) In order for w_0 to be an optimal primal solution and α_0 to be an optimal dual solution, it is necessary and sufficient that w_0 is feasible for the primal problem, and

$$\phi(w_0) = L(w_0, b_0, \alpha_0) = \min_w L(w, b, \alpha)$$

2.2.2 Non separable data

Consider the extension to data which are not linearly separable. The aim is to find an optimal hyperplane that minimizes the misclassification error.

Definition 2.2 (Soft margin)**.** The margin of a separating surface is said to be soft if a data point (x_i, y_i) of the training set violates the constraint

$$y_i(w_i^T x_i + b) \geq 1, \quad for \quad i = 1, ..., N$$

It can occur in two different ways: if the data point (x_i, y_i) falls in the region of separation and on the correct side, or if the point (x_i, y_i) falls in the region of separation but on the wrong side. In the first way, the point will be correctly classified, while in the second one it will be misclassified.
Because of these observations, we can introduce in the hyperplane definition non-negative variables $\xi_1, ..., \xi_N$ that measure the deviation of a point from the ideal condition of separability, named *slack variables*, as shown

$$y_i(w_i^T x_i + b) \geq 1 - \xi_i, \quad for \quad i = 1, ..., N \tag{2.17}$$

Thus, if $0 \leq \xi_i \leq 1$, the point falls inside the region of separation on the correct side, while if $\xi_i > 1$ it falls inside the region of separation on the wrong side.
As we said previously, the aim is now to find the optimal hyperplane for which the misclassification error is minimized. One way for doing it is by the minimization of the following functional

$$\phi(\xi) = \sum_{i=1}^{N} I(\xi_i - 1)$$

where I represents an indicator function [2]. However, it can be noted that the

[2] In this context, an indicator function $I(\xi)$ is defined as

$$I(\xi) = \begin{cases} 0, \ if \ \xi \leq 0 \\ 1, \ if \ \xi \geq 0 \end{cases}$$

minimization of $\phi(\xi)$ with respect to ξ is a non-convex optimization problem, and therefore $\phi(\xi)$ is usually approximated with another functional

$$\phi_1(\xi) = \sum_{i=1}^{N} \xi_i$$

Furthermore, the resulting functional to be minimized with respect to w and $\{\xi_i\}_{i=1,...,N}$, under the constraint (2.17), becomes

$$\phi(w, \xi) = \frac{1}{2} w^T w + C \sum_{i=1}^{n} \xi_i \tag{2.18}$$

The parameter C represents a tradeoff between the complexity of the algorithm and the number of non separable data. Most of the time, it has to be determined by the user.
It follows immediately that the linearly separable problem is included in this formulation, considering $\xi_i = 0$, for $i = 1, ..., N$ in (2.17) and (2.18).

We can now formulate the *primal problem* in the case of non separable data: *" Given a training set D=*$\{(x_i, y_i)\}_{i=1,...,N}$*, find the optimum values of the weight vector w and the bias b such that they satisfy the constraints*

$$(i)\ y_i(w^T x_i + b) \geq 1 - \xi_i, \quad for \quad i = 1, ..., N$$
$$(ii)\ \xi_i \geq 0, \quad for \quad i = 1, ..., N$$

and such that w and the slack variables $\{\xi_i\}_{i=1,...,N}$ *minimize the functional cost*

$$\phi(w, \xi) = \frac{1}{2} w^T w + C \sum_{i=1}^{n} \xi_i$$

where C is a positive parameter chosen by the user. "

Applying a similar procedure used in Section 2.2.1 for the linear case, we can state the dual problem for non separable data, making use of the Lagrangian multipliers:

" Given the training set D, find the Lagrangian multipliers $\{\alpha_i\}_{i=1,...,N}$ *that maximize the objective function*

$$Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x_i^T x_j$$

subject to the contraints

$$\sum_{i=1}^{N} \alpha_i y_i = 0, \qquad 0 \leq \alpha_i \leq C$$

where C is a positive parameter chosen by the user. "

Observation 1. Firstly, it can be noted that the dual formulation for non separable data does not make use of the slack variables, therefore it is much similar to the dual formulation seen in Section 2.2.1. The objective function $Q(\alpha)$ to be maximized is the same in both cases, but subject to different constraints: the condition $\alpha_i \geq 0$ for the first problem is substitued by $0 \leq \alpha_i \leq C$ for non separable data. Thus, for non separable data, the constrained optimization problem for detecting the optimum values of (w, b) proceeds in the same way as for separable data.

Observation 2. A different method when dealing with a k-multiclass classification problem ($k > 2$) can be found in [3]. It is based on the simplest intuition about the problem: a multiclass classifier can be seen as a combination of a number of linear discriminant classifiers. There are two main approaches based on this idea. The *one-vs-the-rest* is the most used strategy that constructs k separate SVMs, in which for each class C_i, $i = 1, ..., k$, the elements in the current class C_i are considered as the positive data and the data of the remaining $k - 1$ classes as the negative ones. Therefore, the multiclass classifier is obtained by using $k - 1$ binary classifiers. The main disadvantage of this construction is that there may be some examples assigned to multiple classes simultaneously. A different approach is called the *one-vs-one* classifier, where the positive class has target $+1$, while the negative class has target $\frac{-1}{k-1}$, thus a weight coefficient for the classes is introduced.

2.2.3 Kernels, feature map and feature space

The Support vector machine is also referred as a *kernel* machine. The main idea is that if data in the input space are not linearly separable, the SVM algorithm can map the input space on a higher dimensional one, where it is possible to separate the data. Thus, it can occur that the resulting system becomes too complex and the calculation of the Euclidean distance between each training point and the separating surface becomes too hard. In this case, SVM may introduce kernel functions, which operate in a feature space and calculate only the inner product between images of points in the feature space, instead of the Euclidean

distance. This 'trick' is computationally simple and therefore allows SVM to be used in high dimensional classification problem. We explain in this section the fundamental notions of kernel, feature map, feature space, with reference to [20] and [9].

Definition 2.3 (Kernel)**.** Let X be a non-empty finite dimensional set. A function k: X×X $\to \mathbb{R}$ is called a *kernel* on X if there exists a $\mathbb{R}$-Hilbert space $H = (H, \langle\cdot,\cdot\rangle)$ and a map ϕ: X$\to$ H such that $\forall x, x' \in$ X we have

$$k(x, x') = \langle\phi(x), \phi(x')\rangle$$

where ϕ is called feature map and H is called the feature space of k.

In other words, a kernel is a function that computes the inner product of the images produced in the feature space under the embedding ϕ of two data points x, x' in the input space X.

Observation 3. ϕ and H are not uniquely determined.

Observation 4. We can consider $k(x, x')$ as the ij-th element of a symmetric $N\times N$ matrix K. The matrix K is a nonnegative definite matrix called the *kernel matrix*.

Whenever H is separable, since it has a countable orthonormal basis it follows the isomorphism $H \cong \ell_2$, we have

Proposition 2.2.2 (Series representation of kernel)**.**
Let X be a non-empty set, consider $f_n : X \to \mathbb{C}$, $n \in \mathbb{N}$, such that $\big(f_n(x)\big) \in \ell_2$, $\forall x \in X$. Then

$$k(x, x') := \sum_{i=1}^{\infty} f_n(x)\overline{f_n(x')}, \quad for\ x, x' \in X$$

defines a kernel on X.

Proof. It follows from the definition and from the fact that the scalar product in ℓ_2 is defined as the sum of the series:

$$k(x, x') = \langle f(x), f(x')\rangle_{\ell_2}$$

□

More details on kernels properties can be found in [20].

Observation 5 (Gaussian RBF kernel). An example of real-valued kernel, one of the most used in practice, is the Gaussian RBF (radial basis function) kernel with width γ, defined by

$$k_\gamma(x, x') := exp\bigg(- \frac{\|x - x'\|_2^2}{\gamma^2}\bigg),\ with\ x, x' \in \mathbb{R}^d$$

It can be derived as the restriction to $\mathbb{R}^d$ of the more general complex kernel defined in $\mathbb{C}^d$:

$$k_{\gamma,\mathbb{C}^d}(z, z') := exp\bigg(- \gamma^{-2} \sum_{j=1}^{d} (z_j - \bar{z_j})^2 \bigg)$$

The introduction of these notions allow us to derive the equation of the optimal hyperplane using a kernel function. The basic motivation behind this approach is due to *Cover's theorem*, which can be formulated in the following form [9]

Theorem 2.2.3 (Cover's Theorem)**.**
A complex pattern-classification problem, cast in a high-dimensional space nonlinearly, is more likely to be linearly separable than in a low-dimensional space, provided that the space is not densely populated.

This result plays a central role when we have non separable data, and we want to classify them with a Support vector machine in a different manner from what exposed in Observation 2.

Consider an input vector x in a finite dimensional input space X. Let $\{\phi_i\}_{i=1}^{\infty}$ be an infinte set of feature map, defined by X to a Hilbert space H. We may define a separating hyperplane according to the formula

$$\sum_{i=1}^{\infty} w_i \phi_i(x) = 0$$

where $\{w_i\}_{i=1}^{\infty}$ defines an infinite set of weights that tranform the feature space in the output space. In a more compact way, this can be written as

$$w^T \varphi(x) = 0 \tag{2.19}$$

where $\varphi(x)$ is the feature vector and w is the corresponding weight vector.
Our aim is now to find a separating hyperplane in the feature space H, as we have done in Section 2.2.1 and 2.2.2.
From Eq. (2.15), in this particular context it becomes

$$w = \sum_{i=1}^{N_s} \alpha_i y_i \varphi(x_i) = 0$$

with N_s the number of support vectors. Substituing Eq. (2.19), we can write the separating hyperplane in the output space as follows

$$\sum_{i=1}^{N_s} \alpha_i y_i \varphi^T(x_i)\varphi(x) = 0 \underset{k(x_i,x)=\varphi^T(x_i)\varphi(x)}{\Longrightarrow} \sum_{i=1}^{N_s} \alpha_i y_i k(x_i, x) = 0 \tag{2.20}$$

Because of the last equation, it can be observed the reason why Support vector machine is often referred to be a kernel machine. In fact, from Eq. (2.20), it follows that we have never to calculate the weight vector w_0, because specifying the kernel is sufficient. This also motivates the reason why Eq. (2.20) is called the *kernel-trick*. An important observation, in particular for applications, is that whenever the feature space is defined as an infinite dimensional space, the Eq. (2.20) defining the optimal hyperplane consists of a linear finite sum of terms, in particular equal to the number of support vectors.

2.2.4 SVM in EMG pattern recognition

In literature, we can find a lot of papers in which comparisons between SVM and other classification algorithms are proposed. For istance, in [16] three different classifiers are tested on twenty able-bodied subjects and one transradial amputee, namely Linear discriminant analysis (LDA), Support vector machine and Artificial neural networks (ANN). A Gaussian kernel has been considered. The classifiers' performances are compared, both with Principal component analysis and Common spatial pattern technique, revealing that Artificial neural networks performs higher than SVM or LDA do. Therefore, this paper shows that in a pattern recognition based system control, analyzing the EMG in the time domain with that specific features, the best classification accuracy is obtained by a neural networks algorithm rather than a support vector machine. Thus, SVM is not universally the best classifier for EMG signals classification.
Another interesting study based on SVM as EMG classifiers can be found in [4], where EMG signals and force signals are used to train Support vector machine with Radial basis function kernel. The experimental subjects are three hand amputees; five surface electrodes are used, with the goal of discriminating phantom limb postures and approximating the required force. A supervised learning strategy is followed, in which the recordings are made according to the modalities of teacher imitation, bilateral action and mirror box. In particular, the hyperparameters γ (width of the Gaussian kernel) and C are found by a logarithmic grid search. Each subject reveals an highest performance in a specific recording

modality, but on average their best performances with the use of a SVM classification algorithm are between $92.64 \pm 0.74\%$ and $95.74 \pm 1.15\%$. Furthermore, the low percentage of support vectors found in the best models indicate that the problem is not difficult from the point of view of machine learning.
A different experiment concerning the use of SVM is [1], with the particular goal of analyzing the best placement of four sensors and the variability of training data along different days with reference to different positions of arm and forearm during the recordings. The results show that the use of four correctly placed electrodes and a slight signal pre-processing can give good result of classification. Therefore, focusing on the correct experimental procedure, high classification accuracies can be obtained with a SVM classifier using four electrodes, and thus is not necessary to look for more complex algorithms with more electrodes.
A further development based on [1] is presented in [17], where an hybrid EMG classifier is proposed by combining a Support vector machine and an Hidden markov model (HMM). In particular, HMM is used to distinguish between steady-state signals from transient one, and then SVM is used to classify the EMG signal during steady-state. The reason why HMM is introduced is to allow the classification of transients, not made possible by a time depending algorithm. In conclusion, the results of the experiment show that an increase on the gesture classification higher than 12% is reached by the hybrid approach.

2.3 Clustering

Clustering is a method of exploratory data analysis based on the grouping of data according to a certain notion of similarity between them. Its aim is to construct groups of data (*clusters*), such that data in the same cluster share similar characteristics, while data in different clusters are dissimilar. It is a technique used in many fields, as statistical data analysis, machine learning, pattern recognition and data compression.
One approach can be a statistical method, based on the assumption that there is a probabilistic model that generates the data points, while one of great interest is the *similarity-based* method. It defines a similarity function between pairs of data points and formulates a criterion based on it, so that the clustering method must optimize. The central point for clustering optimality is therefore the definition of a 'good' similarity function.
In particular, there exist two main categories of clustering: *partitioning methods* and *hierarchical methods*. The first construct k clusters such that for each cluster

there exists at least one element, and each element belongs to one and only one group. These requirements may be summarized as follows

$$(i)\ \forall C_i,\ \exists x_j \in C_i,\ \forall i = 1, ..., k$$
$$(ii)\ \forall i, j \in \{1, ..., k\},\ C_i \cap C_j = \emptyset$$

where C_i is the i-th cluster, $X = \{x_1, ..., x_N\}$ is the set of observations to be grouped in k clusters.

The hierarchical methods instead build a hierarchy of clusters, on the basis of the directions followed in the construction: there is an agglomerative approach, if a bottom-up strategy is followed, or a divisive approach, if the top down strategy is used. In general, the choice of the kind of clustering depends on the structure of data available and on the specific purpose of the study.
In this Section we focus on one of the most used partitioning methods, namely the K-means clustering, which is more suitable dealing with large datasets.

2.3.1 K-means algorithm

Let $Y = \{x_n\}_{n=1,...,N}$ be a set of N realizations/samples of a random variable X in $\mathbb{R}^D$. As first assumption, we consider that the number K of clusters is given by the user.

Definition 2.4 (1-of-K encoder)**.** We call 1-of-K encoder the function $r_{nk} \in \{0, 1\}$ which assigns each data point to one cluster. Equivalentely,

$$r_{nk} = \begin{cases} 1,\ if\ x_n \in C_k \\ 0,\ otherwise \end{cases}$$

with C_k denoting the k-th cluster.

As we said above, the idea of the k-means is to assign both x_i and x_i' to the same cluster if the similarity distance between them is small enough, otherwise to different clusters, and repeat this procedure for all the pairs of data points. In the specific, the k-means is characterized by the use of the squared Euclidean distance as similarity distance, therefore, denoting μ_k the prototype of cluster C_k, we may introduce an objective function as

$$J(r_{nk}, \mu_k) := \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} ||x_n - \mu_k||^2 \tag{2.21}$$

It is a summation over all the data available of the sum of the squared Euclidean distances between each data point to its prototype μ_k. Our goal is to find

the values of $\{r_{nk}\}$ and $\{\mu_k\}$ that minimize Eq. (2.21). It can be obtained with an iterative procedure where, for each step, two optimization problems have to be solved, one concerned $\{r_{nk}\}$ and the other concerned $\{\mu_k\}$. Given some initial values for the μ_k, the algorithm can be summarized as in the following table:

K-means algorithm
Repeat until convergence:
1. Fixed μ_k, solve $$\min_{r_{nk}} J(r_{nk}, \mu_k)$$
2. Fixed r_{nk}, solve $$\min_{\mu_k} J(r_{nk}, \mu_k)$$

Table 2.1: K-means algorithm iterative scheme

On the one hand, to execute (1) in Table (2.1) we can use the linearity of Eq. (2.21) with respect to r_{nk}. We can optimize for each n separately, assigning each x_n to the closest cluster center, according with the formula

$$r_{nk} = \begin{cases} 1, \ if \ k = arg\min_j ||x_n - \mu_j||^2 \\ 0, \ otherwise \end{cases} \tag{2.22}$$

On the other hand, as Eq. (2.21) is quadratic with respect to μ_k, we can differentiate and pose the result equal to zero, as follows

$$\frac{\partial J(r_{nk}, \mu_k)}{\partial \mu_k} = 2\sum_{n=1}^{N} r_{nk}(x_n - \mu_k) = 0 \Leftrightarrow \mu_k = \frac{\sum_n r_{nk} x_n}{\sum_n r_{nk}} \tag{2.23}$$

It can be observed that the denominator of Eq. (2.23) is equal to the total number of data assigned to cluster k, because of its definition. Therefore Eq. (2.23) is an estimate of the mean value of data belonging to the k-th cluster.
Keeping in mind this observation, the functional cost in Eq. (2.21) can be seen as the sum over the total number of clusters of the squared Euclidean distance between each point and the estimate of the mean value of the cluster. In other words, it is a linear summation of the estimates of the variances associated to each cluster

$$J(r_{nk}, \mu_k) = \sum_{k=1}^{K} \sigma_k^2$$

The two steps illustrated in Table (2.1) are reapeated until the total number of iterations is reached or when no more assignment is possible. Usually, the algorithm used for solving these optimization problems is the gradient descent, but it may occur that J converges to local minima rather than global minima. An important observation may be done about the choice of the initial prototypes μ_k. If they are deliberately chosen by the user, the algorithm may take several steps to reach convergence. It is suggested in literature that the best choice for improving the running time and the quality of the final solution should be a random subset of k data points, as implemented in the kmeans algorithm for Matlab [**?**].

Observation 6. In this study, we use the K-means algorithm as a supervised method for individuating exactly a number of clusters equals to the number of gestures an artificial device should have to reproduce. Therefore, we do not mention the techniques most used for the choice of the number of clusters.

2.4 Spectral clustering

Recently, spectral methods have become popular methods in the similarity-based approach. These methods find theoretical motivation in the field of graph theory. The growing success of such approach is mainly due to the simple implementation, because spectral methods are formulated as eigenvalues/vectors problems, thus requiring standard linear algebra methods.
In this section we recall some basics of graph theory, then we formulate the similarity graph approach. Following [12], we also discuss a random walks view of spectral clustering and connections with graph cut point of view.

2.4.1 Similarity graphs

We introduce some basic concepts of graph theory and discuss different type of graphs that can be constructed.

Notation 1 (Basic concepts of graph theory). A graph is defined as a pair $G = (V, E)$, with $|V| = n$. We call $V = \{v_1, ..., v_n\}$ the set of vertices and E the set of edges.
A similarity graph is a pair $G = (V, E)$, together with a similarity matrix (s_{ij}). Two vertices v_i, v_j are connected if the similarity s_{ij} between them is $s_{ij} \geq 0$ or $s_{ij} \geq \ threshold$; therefore the edge between v_i and v_j will be weighted by s_{ij}.

Let X be the set of data point x_i, with $i \in I$. Let $G = (V, E)$ be an undirected graph with vertex set $V = \{v_1, ..., v_n\}$, which means that edges have no orientation, and each vertex v_i represents a data point x_i. We assume that G is weighted, meaning that two vertices v_i, v_j are connected by an edge weighted by a non-negative weight w_{ij}. If v_i and v_j are not connected, $w_{ij} = 0$.
We call adjacency matrix of the graph G the matrix with $n \times n$ elements, defined by $W = (w_{ij})_{i,j=1,...,n}$.
As G is undirected, we require that $w_{ij} = w_{ji}$.
The degree of a vertex $v_i \in V$ is defined as

$$d_i = \sum_{j=1}^{n} w_{ij}$$

We define degree matrix D the diagonal $n \times n$ matrix with elements $d_1, ..., d_n$ on the diagonal.

Given a set of data points $x_1, ..., x_n$ with pairwise similarities s_{ij}, there exist different methods for constructing a graph. The aim is, as we noted above, to create graphs where neighborhood data points are grouped on the base of similar properties. We report three main approaches:

1. *The $\varepsilon-$neighborhood graph*: for every pairwise verteces v_i, v_j, we connect them if $s_{ij} \leq \varepsilon$. This construction makes the resulting graph unweighted: as connected points have at most distances equal to ε, considering weighted edges do not increase informations on the graph.

2. *K-nearest neighborhood graph*: the idea is to connect v_i with v_j if v_j belongs to the k-nearest neighborhood of v_i and vice versa. If the edges' orientation is ignored, the k-nearest neighbor graph is obtained; otherwise, it is called mutual k-nearest neighbor graph.

3. *The fully connected graph*: all the points are connected with each other with positive similarity and all edges are weighted by s_{ij}. A common example for similarity function is the Gaussian similarity function

$$s(x_i, x_j) = exp\left(- \frac{\|x_i - x_j\|^2}{2\sigma^2} \right)$$

where the parameter σ plays an analogous role of the ε in the ε-neighborhood graph.

2.4.2 Graph Laplacians

The fundamental tool in spectral clustering is the graph Laplacian. There exist different ways of defining it and its apllication in graph theory are discussed below.

In the following we assume that $G = (V, E)$ is an undirected, weighted graph with adjacency matrix $W = (w_{ij})$, $w_{ij} \geq 0$. With the statement 'the first k eigenvectors' we will refer to the eigenvectors corresponding to the k smallest eigenvalues.

Definition 2.5 (Unnormalized graph Laplacians)**.**
Given D the degree matrix and W the adjacency matrix of a graph G, the unnormalized graph Laplacian matrix is defined as

$$L = D - W$$

Proposition 2.4.1 (Properties of L)**.**
The matrix L satisfies the following properties:

(i) $\forall f \in \mathbb{R}^n$*, denoting by* f' *its transpose, we have*

$$f'Lf = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2$$

(ii) L is symmetric and positive semi-definite

(iii) The smallest eigenvalue of L is 0, the corresponding eigenvector is the costant one vector 1

(iv) L has non-negative, real valued eigenvalues, $0 = \lambda_1 \leq ... \leq \lambda_n$*.*

Proof. (i) Remembering the definitions, we have

$$\begin{aligned} f'Lf = f'Df - f'Wf = \sum_{i=1}^{n} d_i f_i^2 - \sum_{i,j=1}^{n} f_i f_j w_{ij} = \\ = \frac{1}{2}\Big(\sum_{i=1}^{n} d_i f_i^2 - 2\sum_{i,j=1}^{n} f_i f_j w_{ij} + \sum_{j=1}^{n} d_j f_j^2\Big) = \\ = \frac{1}{2}\Big(\sum_{i,j=1}^{n} w_{ij} f_i^2 - 2\sum_{i,j=1}^{n} f_i f_j w_{ij} + \sum_{i,j=1}^{n} w_{ij} f_j^2\Big) = \\ = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2. \end{aligned}$$

(ii) L is symmetric: in fact, because we have assumed G as an undirected graph, that is $w_{ij} = w_{ji}$, it follows that both W and D are symmetric. The positive semi-definiteness results from (i), because $f'Lf \geq 0$, $\forall f \in \mathbb{R}^n$.

(iii) It follows immediately from the definition and (i), (ii).

(iv) It follows from (i) and (iii). □

Unnormalized graph Laplacians together with their eigenvalues and eigenvectors are able to describe many properties of graphs. There is an important result that allows a better comprehension of spectral clustering, as shown in the following

Proposition 2.4.2 (Number of connected components)**.**
Let G be an undirected graph with weights $w_{ij} \geq 0$. Then, the multiplicity k of the eigenvalue 0 of L equals the number of connected components $A_1, ..., A_k$ in the graph. The eigenspace of eigenvalue 0 is spanned by the indicator vectors $1_{A_1}, ..., 1_{A_k}$ of those components.

For a better clarification of the matrix construction and in particular on the Laplacian matrix structure, see the example below.

Example 1. Consider the graph in Figure (2.3),

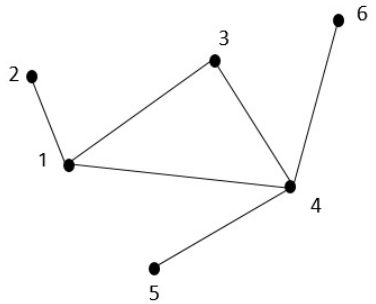

Figure 2.3: Example: undirected graph G=(V,E), with $|V| = 6$

Let construct the degree matrix D and the adjacency matrix W:

$$D = \begin{pmatrix} 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad W = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Therefore,

$$L = D - W = \begin{pmatrix} 3 & -1 & -1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 2 & -1 & 0 & 0 \\ -1 & 0 & -1 & 4 & -1 & -1 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \end{pmatrix}$$

It directly follows that L is symmetric and positive semi-definite.

Definition 2.6 (Normalized graph Laplacians)**.** There are two different matrices called normalized graph Laplacians. They are

$$L_{sym} = D^{-1/2} L D^{-1/2} = D^{-1/2}(D - W)D^{-1/2} = I - D^{-1/2} W D^{-1/2}$$

$$L_{rw} = D^{-1} L = D^{-1}(D - W) = I - D^{-1} W$$

These notations are useful to remind the symmetry property of L_{sym}, and the connection to random walks for L_{rw}.
As in the case of unnormalized graph Laplacians, the main properties of L_{sym} and L_{rw} are summarized in the following two Propositions:

Proposition 2.4.3 (Properties of L_{sym} and L_{rw})**.**
The normalized graph Laplacians satisfy the following properties:

(i) $\forall f \in \mathbb{R}^n$,

$$f' L_{sym} f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} \left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \right)^2$$

(ii) λ *is an eigenvalue of* L_{rw} *with eigenvector* u *if and only if* λ *is an eigenvalue of* L_{sym} *with eigenvector* $w = D^{1/2}u$

(iii) λ *is an eigenvalue of* L_{rw} *wih eigenvector* u *if and only if* λ *and* u *solve the generalized eigenproblem* $Lu = \lambda Du$

(iv) 0 is and eigenvalue of L_{rw} with the constant one vector 1 as eigenvector. 0 is an eigenvalue of Lsym with eigenvector $D^{1/2}1$

(v) L_{sym} and L_{rw} are positive semi-definite and have n non-negative real-valued eigenvalues $0 = \lambda_1 \leq ... \leq \lambda_n$.

Proof. (i) It can be proved in the same way as (i) in Proposition (2.4.1).

(ii) Let consider the eigenvalue equation $L_{sym}w = \lambda w$. From the hypothesis $w = D^{1/2}u$, applying $D^{-1/2}$ on the left, we obtain

$$D^{-1/2}L_{sym}w = D^{-1/2}\lambda w \Leftrightarrow D^{-1/2}L_{sym}w = \lambda u \Leftrightarrow D^{-1/2}D^{-1/2}LD^{-1/2}w = \lambda u$$
$$\Leftrightarrow D^{-1}LD^{-1/2}w = \lambda u \Leftrightarrow L_{rw}u = \lambda u.$$

(iii) It follows immediately that

$$L_{rw}u = \lambda u \Leftrightarrow DL_{rw}u = D\lambda u \Leftrightarrow Lu = \lambda Du.$$

(iv) The first statement is obvious from (iii), because $L_{rw}1 = 0$. The second statement follows from (ii).

(v) The semi-definite positiveness of L_{sym} follows from (i), while the same property for L_{rw} follows from (ii). □

Proposition 2.4.4 (Number of connected components)**.**
Let G be an undirected graph with non-negative weights. Then the multiplicity k of the eigenvalue 0 of both L_{rw} and L_{sym} equals the number of connected components $A_1, ..., A_k$ in the graph. For L_{rw}, the eigenspace of 0 is spanned by the indicator vectors 1_{A_i} of those components. For L_{sym}, the eigenspace of 0 is spanned by the vectors $D^{1/2}1_{A_i}$.

2.4.3 Graph cut point of view

Consider data $x_1, ..., x_n$ given in form of a similarity graph, that means that we have their pairwise similarities $s_{ij} = s(x_i, x_j)$, defined by some similarity function which is symmetric and non-negative. The goal of spectral clustering is therefore to obtain a partition of the graph such that points in different clusters are dissimilar from each other, while points in the same cluster are similar to each other. This may be reformulated asking that edges between different groups have low weights, while edges between the same group have high weights.
In this section we explain how spectral clustering can be derived as an approximation to such graph partitioning problems.

Given a similarity graph $G = (V, E)$ with adjacency matrix $W = (w_{ij})$, we introduce some notations:

- if $A, B \subset V$, we define

$$W(A, B) := \sum_{i \in A, j \in B} w_{ij}$$

- measures for the 'size' of a subset $A \subset V$:

$$\begin{aligned} |A| &:= \textit{the number of vertices in } A \\ vol(A) &:= \sum_{i \in A} d_i \end{aligned}$$

- if $A \subset V$, we denote $\bar{A} = V \setminus A$ as the complemet of A. The set of edges between A and $\bar{A}$ is called a *cut.*

One of the most easy and direct way for constructing a partition of the graph is to solve the *mincut problem*: for a given number k of subsets $A_1, ..., A_k$, the mincut approach consists on minimize

$$cut(A_1, ..., A_k) := \frac{1}{2} \sum_{i=1}^{k} W(A_i, \bar{A}_i) \tag{2.24}$$

However, most of the time this approach does not give good partitions of the graph. This is due to the fact that the minimization of Eq. (2.24) usually separates one single vertex from the rest of the graph, and this is not an acceptable result because clustering's aim is to define large regions of points. To overcome this problem we can ask that the subsets $A_1, ..., A_k$ have acceptable size. For doing so, objective functions that take into account a measure of the subsets $A_i, i = 1, .., k$ have been introduced. The two common functions used are *RatioCut* and *NCut*, defined as above

$$\begin{aligned} RatioCut(A_1, ..., A_k) &:= \frac{1}{2} \sum_{i=1}^{k} \frac{W(A_i, \bar{A}_i)}{|A_i|} \overset{(2.21)}{=} \sum_{i=1}^{k} \frac{cut(A_i, \bar{A}_i)}{|A_i|} \\ NCut(A_1, ..., A_k) &:= \frac{1}{2} \sum_{i=1}^{k} \frac{W(A_i, \bar{A}_i)}{vol(A_i)} \overset{(2.21)}{=} \sum_{i=1}^{k} \frac{cut(A_i, \bar{A}_i)}{vol(A_i)} \end{aligned} \tag{2.25}$$

It can be observed that both objective functions try to create balances clusters, in terms of number of vertices (RatioCut) or edge weights (NCut).
Unfortunately, the addition of balancing functions increase the difficulty of the

minimization problem, in the first time somewhat simple. Spectral clustering is a way to solve relaxed versions of those problems.

We now focus on the approximation of the NCut algorithm, which is deeply connected with random walks on graphs.

The Normalized cut criterion and algorithm

The NCut algorithm was introduced in [19] as a method for solving the mincut problem using the Laplacian matrix as an eigenvalues/vectors problem. For the shake of simplicity, consider only the case of k=2 clusters (to obtain more than two part, procede recursively). We define the cluster indicator vector f by

$$f_i = \begin{cases} \sqrt{\frac{vol(\bar{A})}{vol(A)}}, \ if \ v_i \in A \\ -\sqrt{\frac{vol(\bar{A})}{vol(A)}}, \ if \ v_i \in \bar{A} \end{cases}$$

From this assumption, it follows that $(Df')1 = 0$, $f'Df = vol(V)$, $f'Lf = vol(V)Ncut(A, \bar{A})$. The minimization problem for Ncut can be rewritten as

$$\min_A f'Lf \ s.t. \ f \ as \ above, \ Df \perp 1, \ f'Df = vol(V)$$

Relaxing the problem allowing $f \in \mathbb{R}^n$ and substituing $g := D^{1/2}f$, we obtain

$$\min_{g \in \mathbb{R}^n} g'D^{-1/2}LD^{-1/2}g \ s.t. \ g \perp D^{1/2}1, \ ||g|| = vol(V)$$

Recalling the definition of L_{sym}, from Rayleigh-Ritz theorem, it follows that the solution g is given by the second eigenvector of L_{sym}. In conclusion, re-substituing and recalling the definition of L_{rw}, it follows tha f is the second eigenvector of the generalized eigevalues/vectors problem $Lx = \lambda Dx$. Then f is used to bipartition [3] the graph.

2.4.4 Random walks point of view

A different approach to explain spectral clustering is based on random walks on the similarity graph. In this terms, spectral clustering can be considered as a method that tries to find a partition of the graph such that the random walk stays long within the same cluster and jumps from one cluster to another.

Definition 2.7 (Random walk on a graph)**.** Given a graph G, a random walk on G is a stochastic process $\left(X_t\right)_{t \in \mathbb{N}}$ which randomly jumps from vertex to vertex.

[3] A bi-partite graph is a graph whose vertices can be divided into two disjoint sets A,B such that every edge connects a vertex of A to one in B.

The transition probability of jumping in one step from vertex v_i to vertex v_j is proportional to the weight w_{ij}, given by:

$$p_{ij} := \frac{w_{ij}}{d_i}$$

We define the transition matrix of the random walk as

$$P = (p_{ij})_{i,j=1,..,n}, \quad P = D^{-1}W$$

where D is the degree matrix and W is the adjacency matrix. P can be considered as the stochastic matrix obtained from W by 'normalizing' with D.

Observation 7 (Stationary distribution of the random walk). Let G=(V,E) be a connected and non bi-partite graph with $|V| = n$, then the random walk has a unique stationary distribution

$$\pi = (\pi_1, ..., \pi_n)', \quad where \quad \pi_i = \frac{d_i}{vol(V)}$$

In fact, recalling that

$$p_{ij} = \frac{w_{ij}}{d_i}; \quad vol(V) = \sum_{i=1}^{n} d_i; \quad d_i = \sum_{j=1}^{n} w_{ij}$$

and that $\pi = (\pi_1, ..., \pi_n)$ is a discrete probability distribution such that

$$\forall i = 1, ..., n, \ \pi_i \geq 0; \quad \sum_{i=1}^{n} \pi_i = 1; \quad \forall j = 1, ..., n, \ \sum_{i=1}^{n} \pi_i p_{ij} = \pi_j$$

it follows that

$$\pi_j = \sum_{i=1}^{n} \pi_i p_{ij} = \sum_{i=1}^{n} \pi_i \frac{w_{ij}}{d_i} = \sum_{i=1}^{n} \frac{d_i}{vol(V)} \frac{w_{ij}}{d_i} = \frac{1}{vol(V)} \sum_{i=1}^{n} w_{ij} = \frac{d_j}{vol(V)}.$$

Reminding the Definition (2.5), the relation between the normalized graph Laplacian L_{rw} and a random walk is now clear: in fact, $L_{rw} = I - D^{-1}W = I - P$. Then, λ is an eigenvalue of L_{rw} with eigenvector u if and only if $1 - \lambda$ is an eigenvalue of P with the same eigenvector u.

An important result on the relation between graphs and random walks is due to [12], in particular it is shown the equivalence between the spectral problem formulated by the Ncut algorithm and the eigenvalues/vectors of the transition matrix P.

Proposition 2.4.5. *Consider the two eigenvalues/vector problems:*

$$Lx = \lambda Dx \tag{2.26}$$

$$Px = \lambda x \tag{2.27}$$

where (2.26) is the generalized problem solved by NCut algorithm, while (2.24) is the spectral problem for the matrix P. Then,
if $\lambda,\ x$ *are solutions of (2.27), and* $P = D^{-1}W$*, then* $(1-\lambda),\ x$ *are solutions of (2.26).*

We can also reformulate the NCut criterion in terms of transition probabilities:

Proposition 2.4.6. *Let G be a connected and non bi-partite graph. Assume that we run the random walk* $\left(X_t\right)_{t\in\mathbb{N}}$ *starting with* X_0 *in the stationary distribution* π*. For disjoint subsets* $A, B \in V$*, we denote* $P(B|A) := P(X_1 \in B | X_0 \in A)$*. Then*

$$NCut(A, \bar{A}) = P(\bar{A}|A) + P(A|\bar{A})$$

Equivalentely, if the NCut is small for a certain partition $A, \bar{A}$, it means that the probabilities of evading A once the random walk is in it, and of evading its complement, are both small. Therefore, we have determined a partition of the graph such that the random walk tends to remain in the part where it is, according to the aim we stated at the beginning of this section.

2.5 Neural network

Neural networks, also called artificial neural networks, are machine learning algorithms inspired by biological neural networks. The first introduction of neural network as computing machines dates back to 1943, when the first artificial neurons was introduced by McCulloch and Pitts. The main idea behind the functionality of an artificial neural networks is that it is possible to model the way in which the brain performs a particular task as a network of interconnected artificial neurons.
With reference to [9], we briefly give a description of the neuron model, which consists of three basic elements, as shown in Fig. (2.4):

- a set of *synapses*, that are elementary functional and structural units that mediate the interconnections between neurons. Each synapse is characterized by a weight or strenght which can assume both positive and negative

values, corresponding respectively to excitatory and inhibitory connections. In other terms, when a signal x_j at the input of synapse j is connected to neuron k, it is multiplied by the synaptic weight w_{kj};

- an *adder* that executes the summation of input signals weighted by synaptic weights;
- an *activation function* that constraints the amplitude of the output of a neuron.

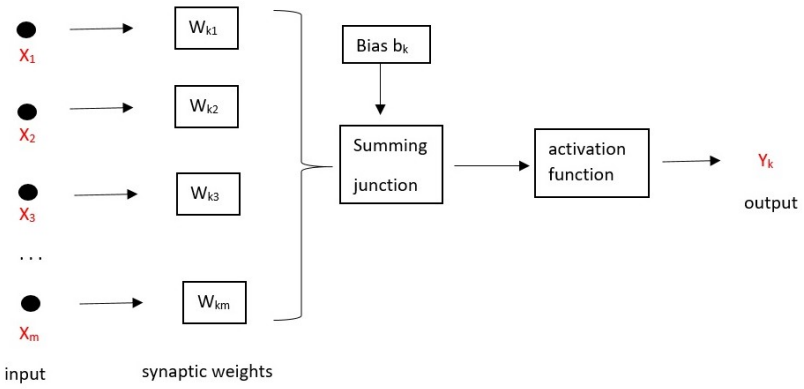

Figure 2.4: Model of an artificial neuron, labelled k

In mathematical terms, the neuron k can be described by the following equation

$$u_k = \sum_{j=1}^{m} w_{kj} x_j \tag{2.28}$$

where $(x_1, ..., x_m)$ is an input signal, $w_{k1}, ..., w_{km}$ are the synaptic weights of neuron k, and u_k is called the linear combiner output. Denoting with φ the activation function, we obtain that the output signal of the neuron k is given by

$$y_k = \varphi(u_k + b_k) \tag{2.29}$$

where b_k represents a possible bias. We can introduce the activation potential $v_k = u_k + b_k$, which allows us to summarize the above equations as follows

$$v_k = \sum_{j=0}^{m} w_{kj} x_j, \qquad y_k = \varphi(v_k) \tag{2.30}$$

where in the formula of v_k we have added a new synaptic weight w_{k0}, corresponding to the input signal $x_0 = +1$; so the weight becomes $w_{k0} = b_k$.

In particular, two basic types of activation function are used: one is the *threshold function* (i), that allows the output to assume the discrete values 0 or 1, while the *sigmoid function* (ii) allows the output to assume continuous values in range between 0 and 1. More explicitly:

$$(i)\ \varphi(v) = \begin{cases} 1, & if \quad v \geq 0 \\ 0, & if \quad v < 0 \end{cases}, \qquad (ii)\ \varphi(v) = \frac{1}{1 + exp(-av)}, \quad a > 0$$

In the first paragraph we describe the perceptron, which is the first algorithmically description of neural networks due to Rosenblatt in 1958. Because it suffers of some limitations, as shown by the exclusive-OR problem, in the second paragraph we present the description of the multi-layer perceptron that allows the classification of nonlinearly separable data.

2.5.1 The perceptron

The perceptron is the simplest form of neural network used for classification of linearly separable data. It consists of one single neuron layer with adjustable synaptic weights and bias, which uses the threshold function as activation function. Its goal is to separate with a plane two collections of input signals belonging to classes C_1 and C_2.
Denoting with n the time-step in applying the perceptron algorithm and denoting with N the total number of signals in the training sample, let us introduce the following notations:

- input vector (also called the impulse): $x(k) = (x_0(k), x_1(k), ..., x_m(k)) = (+1, x_1(k), ..., x_m(k))$, with $k = 1, ..., N$. For every input vector, it is known the belonging of one of the classes C_1, C_2;
- weight vector: $w(n) = (w_0(n), w_1(n), ..., w_m(n)) = (b, w_1(n), ..., w_m(n))$

Substituing these notations in Eq. (2.30), we obtain

$$v(n) = \sum_{j=0}^{m} w_j(n) x_j(k(n)) = w(n)^T x(k(n))$$

Therefore, the decision boundary between classes C_1 (s.t. $\langle w, x \rangle > 0$) and C_2 (s.t. $\langle w, x \rangle < 0$) is defined by the equation $w^T x = 0$, which defines a hyperplane in a m-dimensional space. Thus, the learning rule consists on varying the direction of the decision boundary, that is varying the weight vector, in such a way that positive and negative samples are separated by the boundary itself.

In simple terms, at time step n, if the impulse $x(k(n))$ is correctly classified, the perceptron does not have to adjust the vector weight w; otherwise, a learning rule for adjusting w has to be followed, as summarized below:

$$w(n+1) = w(n) - \eta(n)x(k(n)) \quad if \quad w(n)^T x(k(n)) > 0 \quad and \quad x(k(n)) \in C_2$$

$$w(n+1) = w(n) + \eta(n)x(k(n)) \quad if \quad w(n)^T x(k(n)) < 0 \quad and \quad x(k(n)) \in C_1$$

The parameter $\eta(n)$ is the learning parameter and it controls how much the vector weight has to be adjusted. It takes values in the range $0 < \eta \leq 1$.

An interesting result, known in literature as the *perceptron convergence theorem*, states that for any set of linearly separable data, the perceptron learning rule converges in a finite number of iterations. We present the proof of this theorem in the case of $\eta(n) = 1$. Then, we summarize the algorithm for η a positive constant value.

Theorem 2.5.1 (Fixed-increment convergence theorem for the perceptron - Rosenblatt 1962)**.** *Let H_1 be the subspace of training vectors belonging to class C_1, and let H_2 be the subspace of training vectors belonging to class C_2. Let H_1 and H_2 be linearly separable and at the input presented to the perceptron belong to these subspaces. Then, the perceptron converges after a finite number of iterations.*

Proof. Consider $w(0) = 0$ and suppose that, for n=1,2,..., $w(n)^T x(k(n)) < 0$ with $x(k(n)) \in C_1$, that means that the perceptron does not correctly classify the input $x(k(1)), x(k(2)), ...$ Without loss of generality, consider $\eta(n) = 1$ (this assumption justifies the fixed-increment name).
Because of the assumption, we have $w(n+1) = w(n) + x(k(n))$, for $x(k(n)) \in C_1$. Since $w(0) = 0$, it follows that $w(n+1) = x(k(1)) + x(k(2)) + ... + x(k(n))$. Because we have assumed that the two classes are linearly separable, there must exists a solution w_0 for which $w^T x(k(n)) > 0$ for $x(k(1)), x(k(2)), ..., x(k(n)) \in H_1$.
Now consider w_0 a fixed solution, and define $\alpha := \min_{x(k(n)) \in H_1} w_0^T x(k(n))$. By multiplying w_0^T for the update rule for w, we obtain

$$w_0^T w(n+1) = w_0^T x(k(1)) + ... + w_0^T x(k(n)) \Rightarrow w_0^T w(n+1) \geq n\alpha$$

If we apply the Cauchy-Schwarz inequality to the vectors w_0 and $w(n+1)$, it follows that

$$\|w_0\|^2 \|w(n+1)\|^2 \geq \left(w_0^T w(n+1)\right)^2 \geq n^2\alpha^2 \Rightarrow \|w(n+1)\|^2 \geq \frac{n^2\alpha^2}{\|w_0\|^2}$$

We can rewrite the update rule for w as $w(j+1) = w(j)+x(k(j))$, for j=1,2,...,n, and for $x(k(j)) \in H_1$. Considering the squared Euclidean norm and the hypothesis $w(j)^T x(k(j)) < 0$, it follows that

$$\|w(j+1)\|^2 - \|w(j)\|^2 \leq \|x(k(j))\|^2 \Rightarrow \|w(n+1)\|^2 \leq \sum_{j=1}^{n} \|x(k(j))\|^2 \leq n\beta$$

where $\beta := \max_{x(k(j))\in H_1} ||x(k(j))||^2$. This last inequality shows that the weight vector grows at least linearly with the number of iterations n. Therefore, we have obtained

$$\frac{n^2\alpha^2}{\|w_0\|^2} \leq \|w(n+1)\|^2 \leq n\beta \overset{n\leq n_{max}}{\Longrightarrow} \frac{n_{max}^2\alpha^2}{\|w_0\|^2} = n_{max}\beta$$

where n_{max} is the maximum number of iterations that n can assume, and for which the previous inequality hold with the equality sign. In conclusion, it follows that

$$n_{max} = \frac{\beta\|w_0\|^2}{\alpha^2}$$

Therefore, we have proved that under the hypothesis of $w(0) = 0$, $\eta(n) = 1$ and the assumption of the existance of a solution weight w_0, the learning perceptron rule terminates after at most n_{max} iterations. Thus the proof is conclused. □

In the more general case of $\eta(n)$ constant but not equal to unity, the previous algorithm can be summarized as in Table (2.2).

The perceptron convergence algorithm

Given:

input vector $x(k) = (+1, x_1(k), x_2(k), ..., x_m(k))^T$, with $k = 1, ..., N$

weight vector $w(n) = (b, w_1(n), w_2(n), ..., w_m(n))^T$

Set w(0)=0

For n=1,2,...

1. k(n)=n (mod N)
2. activate the perceptron by applying $x(k(n))$ and desired response $d(k(n))$
3. compute $y(k(n)) = sgn(w(n)^T x(k(n)))$
4. update the weight vector with the *error-correction learning rule*

$$w(n+1) = w(n) + \eta(d(k(n)) - y(k(n)))x(k(n))$$

where $d(n) = \begin{cases} +1, \ if \ x(k(n)) \in C_1 \\ -1, \ if \ x(k(n)) \in C_2 \end{cases}$

Table 2.2: Summary of the perceptron convergence algorithm

Referring to Table (2.2), the vector $y(k(n))$ is called the actual quantized response and η is the learning rate parameter which assumes values $0 < \eta \leq 1$. In addition, the difference $d(k(n)) - y(k(n))$ in the learning rule represents an error signal as the difference between the desired response with the actual response. One of the main difficulties with the perceptron learning rule is that, when data are not linearly separable, then the learning algorithm described in Table (2.2) will never converge.

A famuous example which shows the inefficacy of the perceptron out of linearly separable data is the excluse - OR (XOR) problem reported in the following example:

Example 2 (XOR example). Consider in a two-dimensional space four patterns as shown in Fig. (2.5). The input vectors (0,0) and (1,1) belong to class C_1 while the input vectors (1,0) and (0,1) belong to class C_2. It is evident that there is no linear decision boundary which separates class C_1 from class C_2. This example can be generalized to d-dimensions and it is known as the d-bit parity problem. In the general case, the input set consists of all possible binary input vectors of length d, which are members of class C_1 if the input vector has an even number of 1's, and are members of class C_2 otherwise.

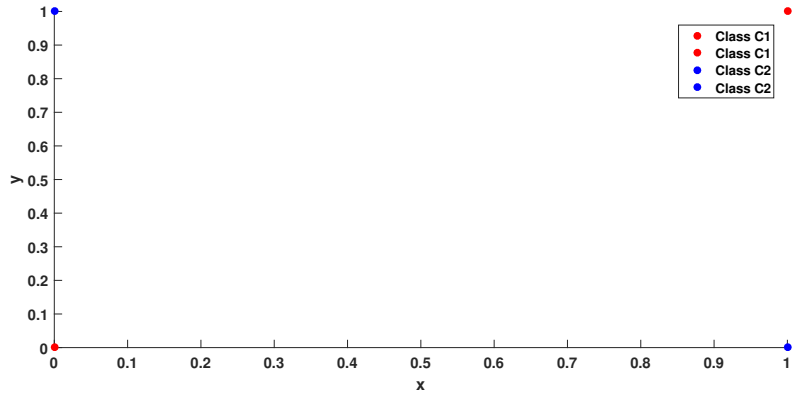

Figure 2.5: Exclusive - OR problem

2.5.2 The multi-layer perceptron

We have seen that the perceptron is a single-layer neural network appliable only when data are linearly separable. To overcome the limitations of this algorithm, we can analyze a more complex neural network structure known as multi-layer perceptron. The main differences between the two typologies of neural networks are due to the number of neurons involved in the network and the different kind of activation function. In fact, the multi-layer perceptron is composed by multiple neurons positioned in multiple layers with a possibly high number of interconnections between them, and it makes use of sigmoidal activation function. Basically, the multi-layer perceptron is individuated by the three following characteristics:

- each neuron in the network is characterized by the use of a nonlinear activation function which is differentiable;
- between the input and output layers there are one or more hidden layers;
- the network has a high connectivity.

A powerful and computationally efficient method for training the multi-layer perceptron is the back-propagation algorithm, which is performed by two following steps. At first there is a *forward* phase, in which the vector weight is fixed and the signal is propagated layer by layer from the input to the output. After, the *backward* phase computes the error signal between the desired network response and the effective response, and propagates it in the opposite direction, layer by layer from the output to the input.

In Fig. (2.6) it is shown the architectural graph of a multi-layer perceptron with one hidden layer. Each neuron is represented as a node in the network and each neuron in any layer is connected to all the neurons in the previous layer (network fully connected). The impulse flow progresses in a forward direction, layer by layer from the input to the ouput layer.

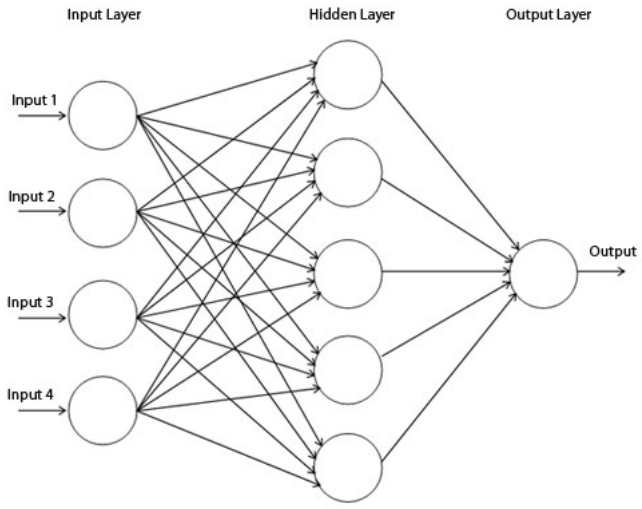

Figure 2.6: Example of multi-layer perceptron fully connected with 4 input neurons, 5 hidden neurons and a single output neuron.

A multi-layer perceptron is characterized also by two different kind of signals which follow different directions through the network. On the one hand there exists the *function signal* which is an input stimulus which comes at the input end of the network and propagates forward through the network, emerging at the output end of the network as an output signal. On the other hand, the *error signal* follows the opposite direction of propagation through the network, since it generates at an output neuron. However, the fundamental difference between the multi-layer perceptron and the perceptron is the existance of hidden layers of neurons. They act as feature detectors, in the sense that they try to find informative characteristics which characterize the training samples. Training the network becomes therefore complicated when we deal with a hidden layer of neurons. We show below how the back-propagation algorithm works when hidden neurons are present in the network.

Preliminary notations

We now introduce some preliminary notations for the description of the back-propagation algorithm.
Let $\mathcal{I} = \{x(k), d(k)\}_{k=1,\dots,N}$ denote the training sample used to train a multi-layer perceptron, with an input layer of neurons, one or more hidden layers and one or more output neurons. With $x(k(n))$ we denote the vector of input stimuli applied to the input layer at time n, while $d(k(n))$ is the corresponding desired-response vector. Considering a stimulus $x(k(n))$ applied to the input layer, let $y_j(k(n))$ denote the function signal produced at the output of neuron j. Therefore, the error signal produced at the output of neuron j is defined by

$$e_j(n) = d_j(n) - y_j(n)$$

We introduce the istantaneous error energy of neuron j as

$$\mathcal{E}_j(n) = \frac{1}{2} e_j^2(n)$$

Let C denote the set of all the neurons in the output layer. Since neuron j is one of the neurons in the output layer, summing all the error-energy contributions $\mathcal{E}_j(n)$, we have

$$\mathcal{E}(n) = \sum_{j \in C} \mathcal{E}_j(n) = \frac{1}{2} \sum_{j \in C} e_j^2(n)$$

In addition, we recall the activation potential associated to neuron j as

$$v_j(k(n)) = \sum_{i=0}^{m} w_{ji}(n) y_i(k(n))$$

and the function signal appearing at the output of neuron j is defined by

$$y_j(k(n)) = \varphi_j(v_j(k(n)))$$

The back-propagation algorithm

The back-propagation algorithm applies a correction Δw_{ji} to the synaptic weight connecting neuron i to neuron j, which is proportional to the partial derivative $\frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)}$. Expanding the partial derivative and recalling the notations above, we have

$$\frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)} = \frac{\partial \mathcal{E}(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(k(n))} \frac{\partial y_j(k(n))}{\partial v_j(k(n))} \frac{\partial v_j(k(n))}{\partial w_{ji}(n)} = -e_j(n) \varphi_j'(v_j(k(n))) y_i(k(n)) \tag{2.31}$$

Therefore, the correction $\Delta w_{ji}(n)$ is defined by

$$\Delta w_{ji}(n) = -\eta \frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)} \tag{2.32}$$

where η is called the learning-rate parameter. Thus, substituing Eq. (2.31) in Eq. (2.32), it follows that

$$\Delta w_{ji}(n) = \eta \delta_j(n) y_i(k(n)) \tag{2.33}$$

where

$$\delta_j(k(n)) = \frac{\partial \mathcal{E}(n)}{\partial v_j(k(n))} = \frac{\partial \mathcal{E}(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(k(n))} \frac{\partial y_j(k(n))}{\partial v_j(k(n))} = e_j(n) \varphi_j'(v_j(k(n)))$$

is called the local gradient. In other terms, the weight correction is defined by multipling the leaning-rate parameter with the local gradient and the input signal. Now, the key factor in the learning rule is the computation of the error signal $e_j(n)$. It follows immediately that we have to treat separately the case of neuron j as an output unit and the case of neuron j as hidden neuron:

- if neuron j is located in the output layer, the error signal $e_j(n)$ is directly computed by the equation $e_j(n) = d_j(n) - y_j(n)$. Once obtained this value, the local gradient may be directly computed and the correction in the vector weight follows immediately by Eq. (2.33);

- if neuron j is a hidden node, the error signal has to be determined recursively. In particular, we have to redefine the local gradient $\delta_j(k(n))$ for j hidden neuron as

$$\delta_j(k(n)) = -\frac{\partial \mathcal{E}(n)}{\partial y_j(k(n))} \frac{\partial y_j(k(n))}{\partial v_j(k(n))} = -\frac{\partial \mathcal{E}(n)}{\partial y_j(k(n))} \varphi_j'(v_j(k(n))) \tag{2.34}$$

  The total error energy can also be rewritten as $\mathcal{E}(n) = \frac{1}{2} \sum_{i \in C} e_i^2(n)$. Now, differentiting $\mathcal{E}(n)$ with respect to $y_j(k(n))$, we have

$$\begin{aligned} \frac{\partial \mathcal{E}(n)}{\partial y_j(k(n))} &= \sum_{i \in C} e_i(n) \frac{\partial e_i(n)}{\partial y_j(k(n))} = \sum_{i \in C} e_i(n) \frac{\partial e_i(n)}{\partial v_i(k(n))} \frac{\partial v_i(k(n))}{\partial y_j(k(n))} = \\ &= -\sum_{i \in C} e_i(n) \varphi_i'(v_i(k(n))) \frac{\partial v_i(k(n))}{\partial y_j(k(n))} = -\sum_{i \in C} e_i(n) \varphi_i'(v_i(k(n))) w_{ij} = \\ &= -\sum_{i \in C} \delta_i(k(n)) w_{ij}(n) \end{aligned} \tag{2.35}$$

Therefore, from Eq. (2.34) we obtain the following *back-propagation formula* for the local gradient $\delta_j(k(n))$:

$$\delta_j(k(n)) = -\frac{\partial \mathcal{E}(n)}{\partial y_j(k(n))}\varphi_j'(v_j(k(n))) = \varphi_j'(v_j(k(n)))\sum_i \delta_i(k(n))w_{ij}(n)$$

In conclusion, if neuron j is a hidden node, the local gradient equals the product of the factor $\varphi_j'(v_j(k(n)))$ and the weighted sum of the local gradients δ_i with $i \in C$, that are the neurons in the next hidden or output layer connected to neuron j.

2.5.3 Neural networks and EMG signals

It is not surprising that neural networks have attracted attention due to their good trainability, adaptability and non-linear separability, in particular in EMG signal analysis.

An interesting application of neural networks in the control of a prosthetic device can be found in [10]. More precisely, artificial neural networks are expected to learn the relationship between the EMG signals and the corresponding movements of a prosthetic hand or arm. The aim of the cited paper is to present the "Artificial body image", which consists on the constructing of the body image by using the automatic learning of neural networks, insted of by the users of the prosthesis. The network thus learns the relationship between the EMG patterns and the intended finger motions, torque or joint angle of fingers. We focus on the finger motion recognition experiment, in which five stationary finger positions are performed and EMG signals are detected by surface electrodes. The detected signals are FFT-analyzed and passed as input to a typical neural network with back-propagation algorithm. The network consists on 10 input, 7 hidden and 5 output neurons, where the output units correspond to the five finger positions. Thirty training data samples are used to train the network, where the right category related to a FFT-analyzed EMG signal is given to the network as the desired response d_j. Then the synaptic weights w_{ji} are corrected by the learning rule with the aim of reducing the error signal between the desired and the actual responses $e_j = d_j - y_j$. During the experimental session, 1'000 training cycles are performed. As a result, 20 out of 30 new EMG patterns are successfully recognized with a recognition rate of 67%. Moreover, if two-channel EMG are used, by adding another electrode on the extensor digitorum, the recognition rate updates up to 86%. In this case a neural network with 20 input, 20 hidden and 5 output neurons is used.

A different study is presented in [16], where it is found out that neural networks are the best choice to classify EMG data. Twenty able-bodied subjects were asked to perform five different hand gestures and EMG signals are recorded by six surface electrode. In this paper, a neural network with 10 hidden and 5 output neurons is applied, with the scale conjugate gradient back-propagation algorithm. An accurate analysis on validation set is done: the 15% of the training dataset is randomly chosen as validation set, and the network is optimized by repeating the training stage until the validation set reaches a classification accuracy higher than 90%. As a result, Artificial neural network (ANN) revels to be the best classifier for the purpose of the article, compared to Support vector machine and Linear discriminant analysis. In fact, with both Principal component analysis (PCA) and Common spatial pattern (CSP) pre-processing techniques, the average classification accuracies reached by the use of ANN are $87,34 \pm 7,3\%$ (PCA with feature vector Root mean square- Willison Amplitude), and $86,62 \pm 7,34\%$ (CSP with feature vector Mean- Root mean square- Willison Amplitude).

Chapter 3

Experiments and results

In this chapter we report the results obtained during the study.
It is not known at all which is the information included in EMG signals and expecially where it is located. Therefore, machine learning algorithms are widely used for pattern classification, as we reviewed in Chapter 2.
The motivation behind our study is to construct a fast and low computational cost classification algorithm, which considers the sensors separately along with a few time features. We list the various experiments that enabled us to reach this proposal. More specifically, we first focus on healthy subjects and show how the analysis in the frequency domain may perform good results of clustering, rather than applied on upper limb amputees. In fact, from the comparison between the results obtained in the frequency domain for both the typologies of patients, we can do some considerations on the frequency behavior of EMG amputees' signals. After this first observation, we find interesting that looking at signals as they are, there are some electrodes which are more activated rather than others, suggesting that by themselves they play a discriminating role in pattern classification. From these visual inspections, we may change the perspective and start considering the electrodes separately. Using five domain features and mainly the Euclidean norm, we compute the reciprocal distances between each pair of acquisitions in what we call distance matrices. Making use of them, we firstly construct a manual-classifier, which reveals a global classification accuracy around the 75%. Since the results obtained seem to be quite promising, we try to automate the feature selection.
In Section 3.1 we show the results obtained on healty subjects in the frequency domain. In Section 3.2, after giving a brief review of bad results on amputee subjects in the frequency domain, it is described an easy and computationally

advantageous approach based on the selection of four pairs of time features and distances. In Section 3.3 we present some possibile future developments of our classification method and problems still open in literature.

3.1 Analysis on healthy subjects

In this Section we present the classification results obtained from three healthy subjects, by working in the frequency domain. Data were acquired at INAIL Centro Protesi in Vigorso di Budrio in June 2016. The subjects were two women and one man, aged between 24 and 28, all with right dominant hand. They were asked to perform five hand postures (np) with their dominant hand: rest, fist, pinch, pointing and pronation of the wrist. Each gesture was repeated 10 times (nc), with a random order in the execution. The subjects wore a silicone bracelet placed around the circumference of the arm, about 5cm below the elbow. Six commercially sEMG electrodes (Ottobock 13E200=50) were placed on the bracelet, with frequency sample Fs chosen to be equal to 800 Hz and recording time T for each gesture equals to two seconds.
At the end of the experiment, the resulting dataset is a matrix of dimension $[T \cdot Fs \cdot np \cdot nc \times M]$, where each row corresponds to an acquisition istant time, while the first column corresponds to the target label (an integer number between 1 and 5) and the remaining six colomns correspond to the index of the sensor.
This preliminary step is only based on the analysis of the Fourier Transform of the EMG signals. This is due to the fact that signal analysis typically concerns the study of the signal in the frequency domain, thus changing the recorded signals from time or spatial domain into the frequency domain. The goal is to indentify the frequency components inside a signal, and try to characterize the signal on the base of its frequency features. We thus make use of the Fast Fourier Transform (FFT) [1] to look for the presence of the most significant oscillations.
After the separation of the dataset into the $np \cdot nc$ acquisitions, we can move from the time domain, where the signals are presented as time measurements, to the frequency domain. For each acquisition, we calculate the FFT, making the following observations: since an EMG signal is a real signal, the Fourier

[1] Given a vector X of length n, $Y = FFT(X)$ implements the discrete Fourier transform, which is defined by

$$Y(k) = \sum_{j=1}^{n} X(j) W_n^{(j-1)(k-1)}$$

where $W_n = exp(-\frac{2\pi i}{n})$

Transform is symmetric, thus we may consider only the first half of the Fourier's coefficients. In addition, once computed the absolute value of the FFT, we may visualize where the frequency band with most significant frequencies is located. In general, considering the six sensors sperately, it may be noted that the most informative content is concentrated in the first 100 coefficients, since we may observe that most of the energy is concentrated up to the 100^{th} coefficient rather than in the following ones. In Fig. (3.1) we present an example which justyfies this assumption.

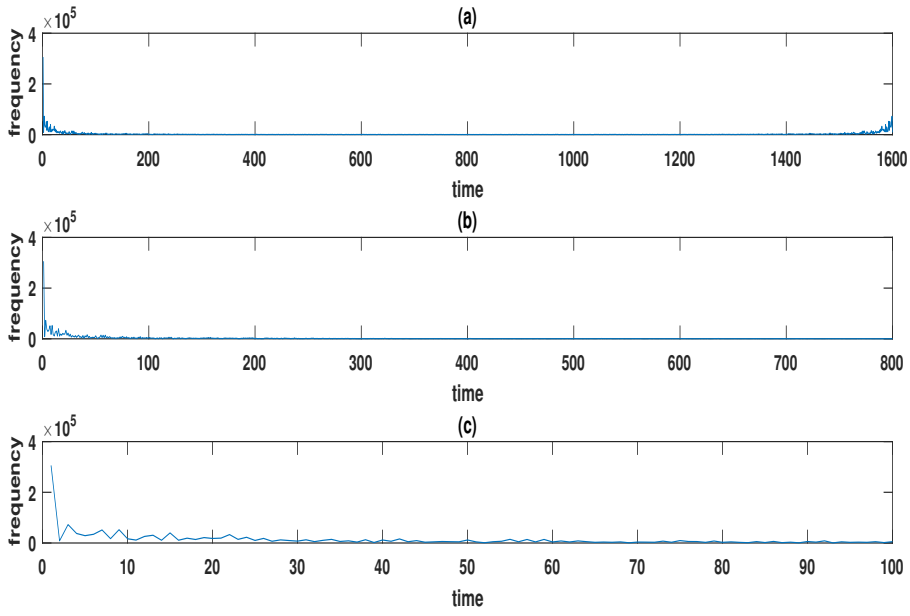

Figure 3.1: Example of simplifications made on the FFT of an EMG signal: in (a) is represented the entire $Y = |FFT(X)|$ of the input signal X, in (b) it is represented the first half of Y, in (c) there are only the first 100 Fourier's coefficient

Making all these simplifications, we obtain, from each sensor, 50 vectors consisting of 100 components in the frequency domain. In a matrix notation, we may construct a global matrix S of 600 rows, as the result of concatenating the features computed for each electrode, and 50 columns.
Once computed this matrix, we can apply the PCA dimensionality reduction algorithm, finding out that three principal components are enough to describe the entire dataset, obtaining uncorrelated variables.
Afterwards, we may apply the Matlab predefined function `kmeans`, with k=5,

(http://it.mathworks.com/help/stats/kmeans.html), for grouping the 50 acquisitions in the frequency domain in order to obtain the corresponding clusters of hand gestures.

As final analysis, to evaluate the goodness of our classification, we may construct two distance matrices based on the reciprocal distances between acquisitions: the main idea is to obtain squared matrices with diagonal blocks, corresponding to the five hand gestures.

The metrics we consider are defined as follows: for two given vectors u, v of N elements, chosen within the totality of acquisitions, we define:

$$\begin{aligned} &(i) \quad d_1(u,v) = ||u-v||_2 = \left(\sum_{j=1}^{N} |u_j - v_j|^2 \right)^{1/2} \\ &(ii) \quad d_2(u,v) = exp\left[- \left(\sqrt{1 - \left| \frac{u \cdot v}{\|u\|_2 \cdot \|v\|_2} \right|^2} \right)^2 \right] \end{aligned} \tag{3.1}$$

The algorithm described above is summarized in Table (3.1).

The frequency based classification algorithm for healthy subjects

Given a matrix D of $m \times n$ dimension, representing the entire registration of EMG signals, where

$m = T \cdot Fs \cdot np \cdot nc$, with T the time of a single acquisition, Fs the sample frequency, np the number of gestures, nc the number of cycles for each gesture;

$n = Label + ns$, where $Label$ is the vector containing the taget labels, ns is the number of electrodes used.

Proceed as follows:

1. separate the $np \cdot nc$ acquisitions from D according to the target label;
2. for every acquisition a_j with $j = 1, ..., np \cdot nc$, calculate $y = |FFT(a_j)|$;
3. reduce the dimension of y by considering the first F Fourier's coefficient;
4. concatenate the ns sensor contributions to construct a global matrix in the frequency domain, denoted by S of $(ns \cdot F) \times (np \cdot nc)$ dimension;
5. solve the PCA problem for reducing the dimensionality of S:
   - let $C = cov(S^T)$
   - solve the eigenvalues/vector problem $[V, D] = \texttt{eig}(C)$
   - sort in descending order both V and D
   - compute the PCs as $w_{ki} = S^T \cdot v_{ki}$, for $k = 1, .., np \cdot nc$, $i = 1, ..., ns \cdot F$
6. apply the np-means to the reduced PCs (Matlab `kmeans` default function);
7. compute the distance matrices with d_1 and d_2 between the first p components of the reduced PCs ($3 \leq p \leq 7$), and evaluate the rate of accuracy of the algorithm.

Table 3.1: Summary of the classification algorithm in the frequency domain for healthy subjects

3.1.1 Healthy patients results

We report the results obtained applying the method described in Table (3.1) to the three healthy subjects: each figure shows in **a** the plot of the first 10 eigenvalues of the covariance matrix C, in **b** the 5-means applied to the first three principal components, in **c** the distance matrix computed with d_1 and in **d** the distance matrix computed with d_2.
With reference to Fig. (3.2), (3.3), (3.4), we report the numerical results concerning our analysis into Table (3.2).

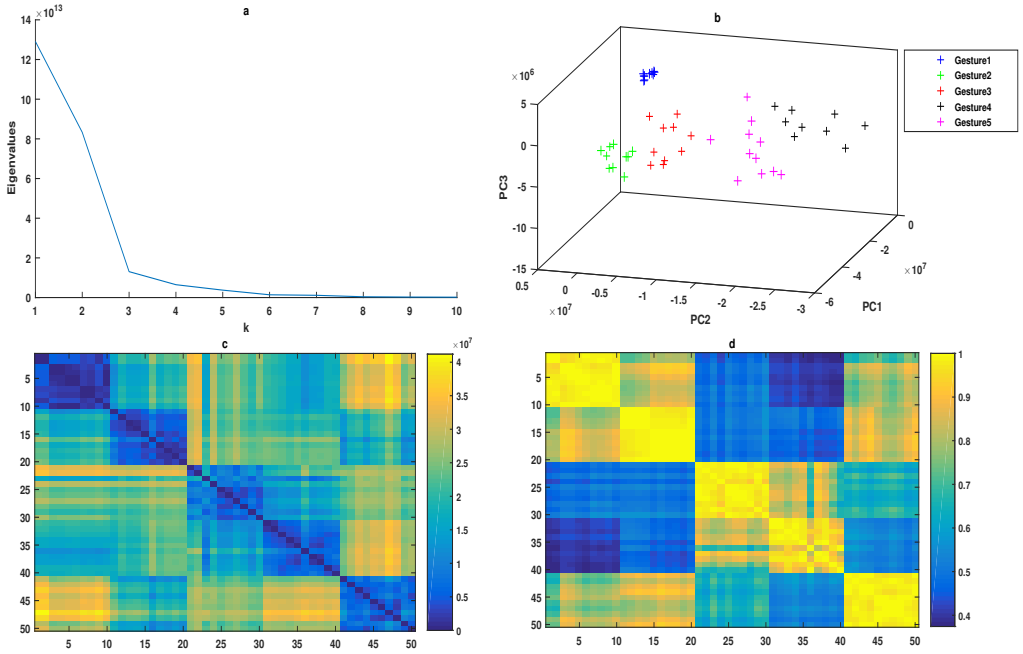

Figure 3.2: Able-bodied subject 1

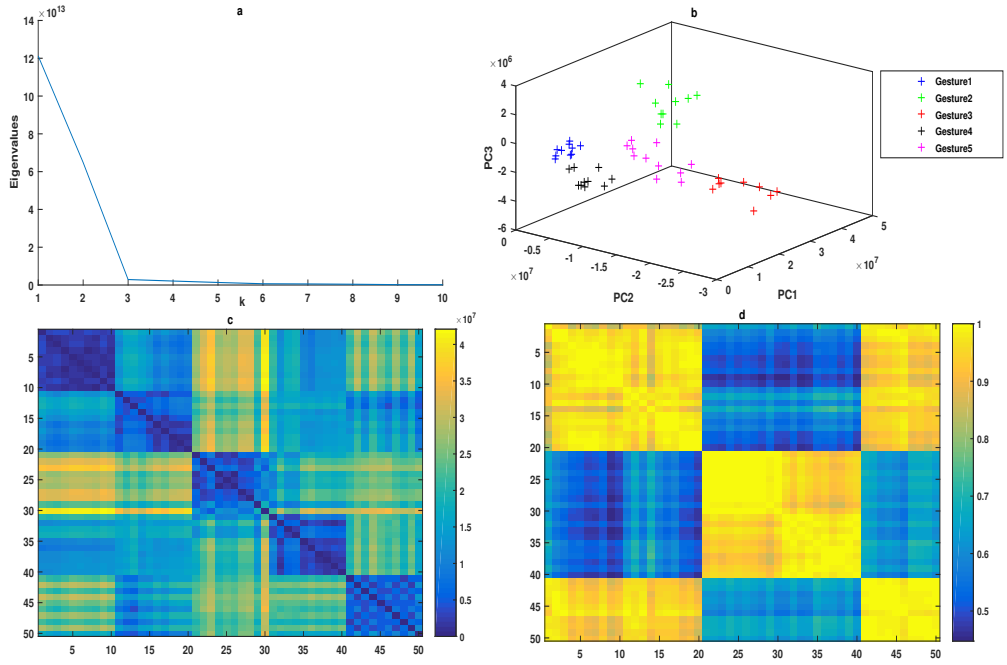

Figure 3.3: Able-bodied subject 2

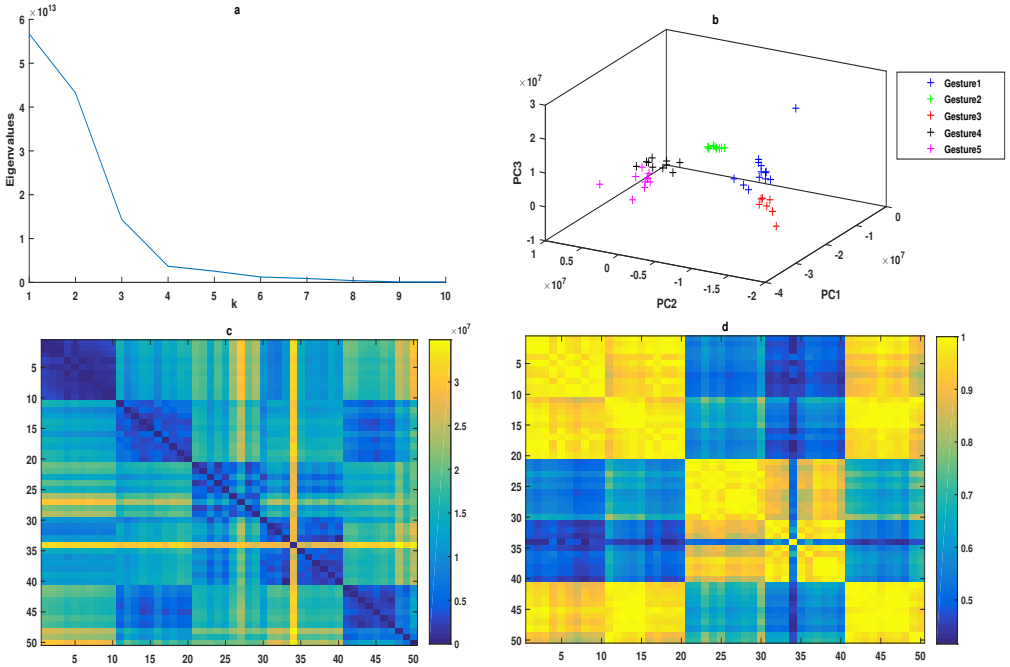

Figure 3.4: Able-bodied subject 3

	n of elements per cluster	execution time
Subject 1	(10,10,10,9,11)	1.78 s
Subject 2	(10,10,9,10,11)	1.35 s
Subject 3	(13,10,7,10,10)	1.05 s

Table 3.2: Numerical results on the classification algorithm: for each subject it is reported the number of elements for each cluster and the execution time

3.1.2 Comments on healthy patients results

In general, the results obtained in this preliminary study are quite promising. Firstly we observe the common decrease of the eigenvalues for all the subjecs: in particular, for subjects 1 and 2, three principal components are enough to project the original variables in order to obtain a good representation of the data, while subject 3 requires one more principal component. In other words, computing the PCA algorithm allows us to obtain a few number of new uncorrelated variables, obtained from an orthogonal rotation of the original axes. We know that the new axes with maximum variance do not garantee to have good features for classification. Therefore, in the reduced space of the first three principal components, we apply the 5-means clustering algorithm. It actually shows five clusters, but analyzing the results reported in Table (3.2), not all the clusters count precisely ten data. This result may be justified by the way in which `kmeans` works, in particular in the way it chooses the initial cluster centroid.
With reference to the distance matrices, it may be noted that they present more pronounced diagonal blocks rather than the others extra-diagonal. Since we have ordered the dataset with respect to the gestures' execution, this means that the distances between the acquisitions of the same class are closer than the ones between different gestures. Furthermore, we may also individuate some acquisitions which are different from the others (for instance in Fig. (3.4) the 34^{th} acquisition is completely different from all the others around it), calling these ones *singular* acquisitions. This detection may be done searching some atypical vertical or horizontal lines in the distance matrices plot. We think that this kind of observation is connected to the identification of wrong acquisitions, during the entire cycle of EMG signal registration.

3.2 Analysis on trans-radial amputee subjects

The analysis reported in Section 5.1 shows that working with healthy patient EMG signals in the frequency domain allow us to find clusters between the data projected into the subspace spanned by the first few principal components. What we expect, based on the previous approach used in [16], as a preliminary step, is the possibility of extending this approach to amputee subjects and check if well separated clusters of data may be detectable. Therefore, we firstly want to analyze if the EMG signals of trans-radial amputees share a similar trend in the Fourier space. Since this does not occur, we change the approach constructing a classifier which considers the sensors separately together with five temporal features.
The subjects of the experiment were 20 trans-radial amputees subjects, patients of INAIL Centro Protesi, aged between 18 and 65, who were already experienced in myolectric control of prosthetic hands. Each of them gave informed consent before performing the experiment. Six commercial sEMG electrodes were used (Ottobock 13E200=50), equidistantly placed on a silicon bracelet, situated on the patient's stump, about 5cm below the elbow. The first sensor was located on the flexor carpi-radialis muscle, while the sixth sensor was located on the brachio-radialis muscle. The experiment consisted on the execution of the same five hand gestures described in Chapter 1, calling the execution of five consecutive hand gestures a cycle. With the support of a monitor interface which depicted, for each gesture, the corresponding image on the display, each execution was recorded for 2 seconds trying to let go the transient signal and considering as much as possible only the steady-state EMG signal. For every patient 10 completed cycles were executed.

3.2.1 Able-bodied vs amputee subjects

Firstly, we compare the decreasing trend of healthy subjects signals to the one of trans-radial amputees, which reveals a deep difference. As first investigation, looking at the Fourier Transform of EMG signals, we can say that the most of information is not concentrated in the first hundreds of coefficients, as it is for healthy subjects. In fact, the Fourier coefficients series does not decay as it does for healthy subjects. An example of this result is shown in Figure (3.5), where we consider the absolute value of the FFT of the EMG signal of a single patient,

centered with respect to the mean [2]. This assumption is supported by a global analysis we have made during this work on the entire dataset. A similar result may be obtained if we look at the first derivative of the signal, computed directly by Definition (3.1). An example of result is depicted in Fig. (3.6).

Definition 3.1 (First derivative of an EMG signal)**.** Let S denote the $L \times ns \times nc$ array of a complete cycle of EMG signal acquisition, where L is the number of times acquisition, ns is the number of electrodes and nc is the number of repetitions for each hand gesture. The first derivative of S is defined as

$$\partial S(l, s, c) = S(l + 1, s, c) - S(l, s, c), \quad for \quad l = 1, ..., L$$

with $s = 1, ..., ns$ fixed sensor and $c = 1, ..., nc$ fixed repetition.

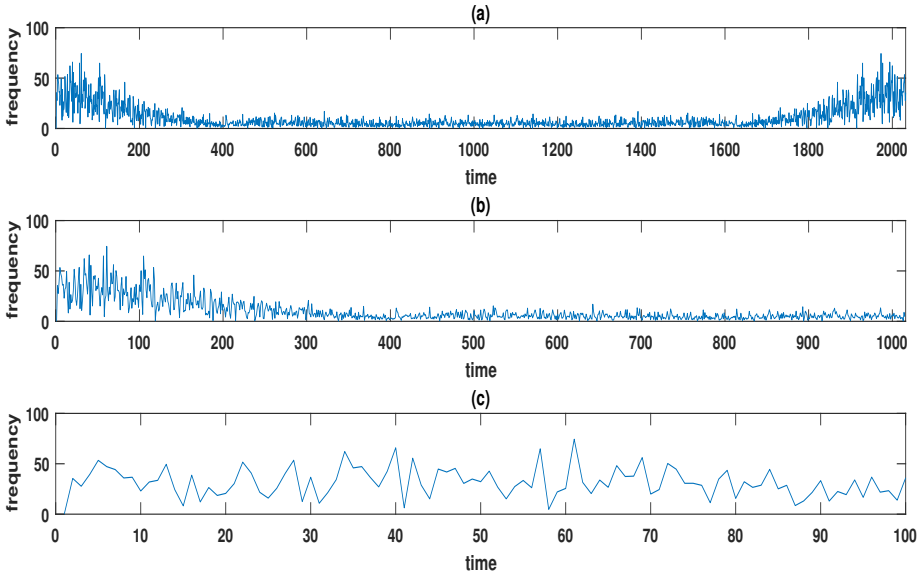

Figure 3.5: Example of the simplifications made on the FFT of an EMG signal for a trans-radial amputee subject: **a** is the absolute value of the FFT of the signal, centered with respect to the mean, **b** is the plot of the first half of Fourier's coefficient, **c** is the plot of the first 100 coefficients

[2]It may be observed that if we do not remove the mean from the data, a first peak at zero may be found. Therefore, in order to obtain clearer plots, we subtract the mean from each registration.

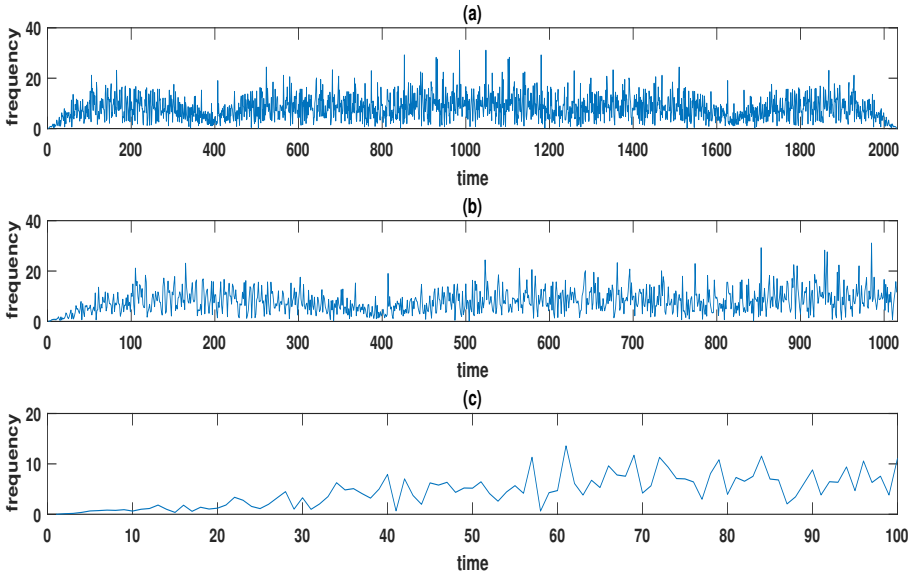

Figure 3.6: Example of the simplifications made on the FFT of the derivative of EMG signal (∂S) for a trans-radial amputee subject: **a** is the absolute value of the FFT of the first derivative of the signal, centered with respect to the mean, **b** is the plot of the first half of Fourier's coefficient, **c** is the plot of the first 100 coefficients

One first observation concerns the non-decay of the Fourier transform of both the signals and the derivatives, and the absence of relevant peaks. This may be due to the non stationarity of the signals. In fact, if the EMG signals were stationary, we would expect the absolute value of the FFT to be similar within the repetitions of the same gestures and different from the other gestures.
One additional reason behind these results of non decay should concern a noise component intrinsic in the signal. To evaluate this possibility, we apply to the entire dataset a noise-reduction algorithm (HaarDenoise), but we do not observe significant differences after the denoising procedure.
Another different reason should be related to the patient's remnant muscle activity. In this context, each subject should be characterized by a different ability in performing the gestures required, with non constant muscle contraction during the totality of the repetitions.

To sum up, we report in Fig. (3.7) the classification results obtained from the frequency-based approach described in Section 1 on one single patient between the 20 trans-radial amputee subjects.

Even if we find a similar decay of the eigenvalues sequence, if we project the data into the subspace spanned by the first principal components and cluster them with 5-means, we obtain a non-equal disposition of the data into the 5 groups (precisely, here we have $n = (10, 4, 4, 19, 3)$). Therefore, the distance matrices depicted in **c** and in **d** are confusing and far from the ones obtained for healthy subjects, which had more pronounced diagonal blocks.
To support these considerations, we have applied the same procedure to the totality of the patients, obtaining always bad results of classification.
Therefore, we consider that changing the perspective can bring more promising results of classification of the five hand gestures.

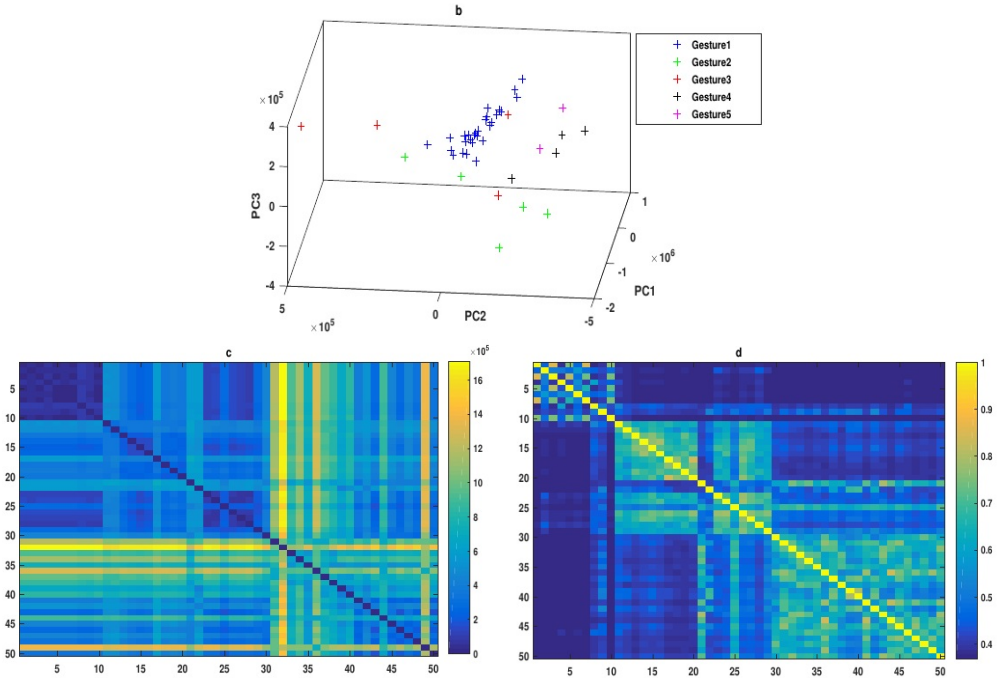

Figure 3.7: Example of frequency-based classification algorithm performed via PCA and 5-means clustering on a trans-radial amputee subject: **b** shows the 5-means applied to the first principal components, **c** is the distance matrix computed via d_1 and **d** is the distance matrix computed via d_2

3.2.2 A new classification approach

We have noted that it is difficult for upper limb amputees to obtain good results of clustering and classification using only the Fourier transform of the EMG signals as for healthy subjects. We describe now a different promising approach.

Up to here, we have followed the procedure described in [16] to train the classifier on healthy subjects and then test it on one amputee. From the Fourier analysis we have seen that this approach can provide bad results. Therefore, we suggest to focus on each patient separately and proceed with the training of a pattern recognition classifier for each subject.
Secondly, the main idea is to change the perspective to analyze the problem: until now, we have assumed the electrodes all together, without a distinct separation between them but concatenating them into a single vector. What we now suggest is to consider the electrodes separately. This fact is basically motivated by two factors: one is the location of the sensor, the other is the specific rule of each electrodes in the detection of different gestures. In fact, it can be quickly observed that for every gesture there are some electrodes activated more than others, and hopefully within the same gesture the same sensors would be active.
For a better clarification, we show in Fig. (3.8) and (3.9) the EMG signal amplitudes for every electrode (reported on the vertical line), in color scale for two different gestures, spread and fist, in nine consecutive repetitions by the same patient. It can be seen that inside the same gesture, some electrodes are more active than others. For instance, in Fig. (3.8) the second electrode works more than the others, in the majority of the repetitions depicted. This suggests that it may characterize the gesture, but the different times at which it is active may represent some differences or errors in the acquisition procedure, since the peaks of the EMG signal are not ordered always in the same way. In other words, we can see that except in the 6^{th} and 9^{th} repetitions, sensor 2 is the one which records the majority of muscle activity for the spread gesture, but at different time instances.

Once observed the behavior of the sensors, we want to construct a classifier which takes the sensors into account separately and computes some basic time domain features. Precisely, we compute the five following time domain features, defined by:

- Normalized EMG signal: it is the amplitude of the signal $x(t)$ normalized by its maximum value for each gesture, between the six electrodes

$$x^N(i,s,j) = \frac{x(i,s,j)}{max(x(j))}$$

where $i = 1, ..., T$ with T the total number of acquisitions istant time, s denotes the number of electrodes used and j denotes the index of acquisition

- Derivative of EMG signal: it represents the 'jumps' of the signal

$$x'(i,s,j) = x(i+1,s,j) - x(i,s,j)$$

- Energy of EMG signal: it corresponds to the gesture's strength of activation

$$En(s,j) = \sum_{i=1}^{T} x(i,s,j)^2$$

- Local mean of EMG signal: it is a variable parameter, computed as the mean value of the EMG signal in 1/10 s

$$M(i,s,j) = E[x(I,s,j)]$$

  where I refers to a time interval of 1/10 s

- Local standard deviation of EMG signal: it is a variable parameter, computed as

$$V(i,s,j) = \sqrt{var(x'(I,s,j))}$$

The metrics used to evaluate the distances between each pair of acquisitions are the 2-norm and the absolute value.

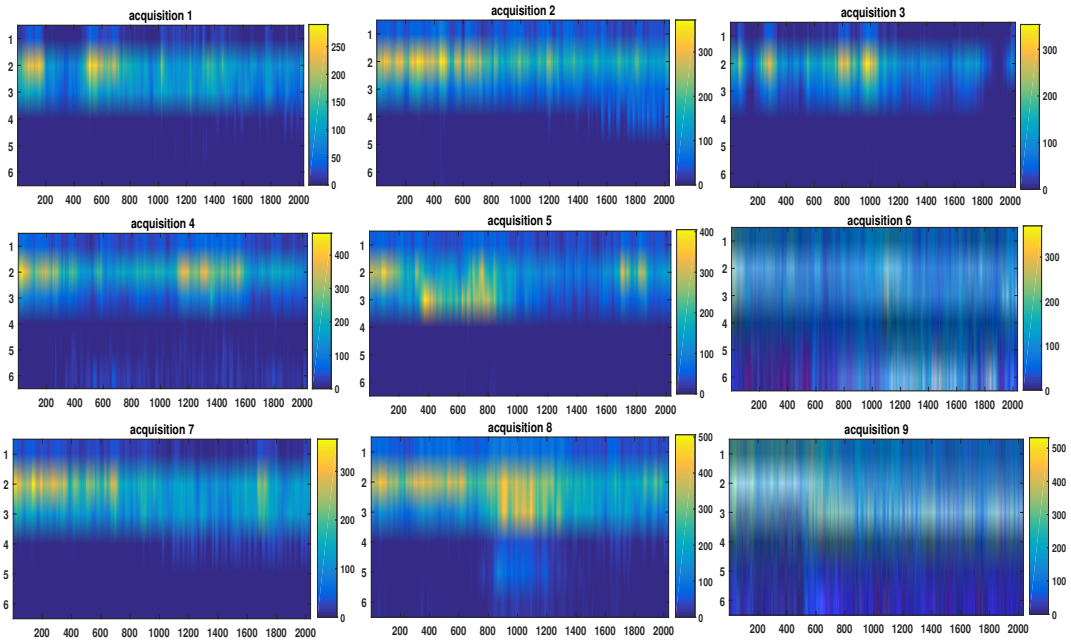

Figure 3.8: EMG signal amplitude of spread hand gesture. The amplitude values are represented in color scale, with respect to the time istances (1-2000) and to the six electrodes

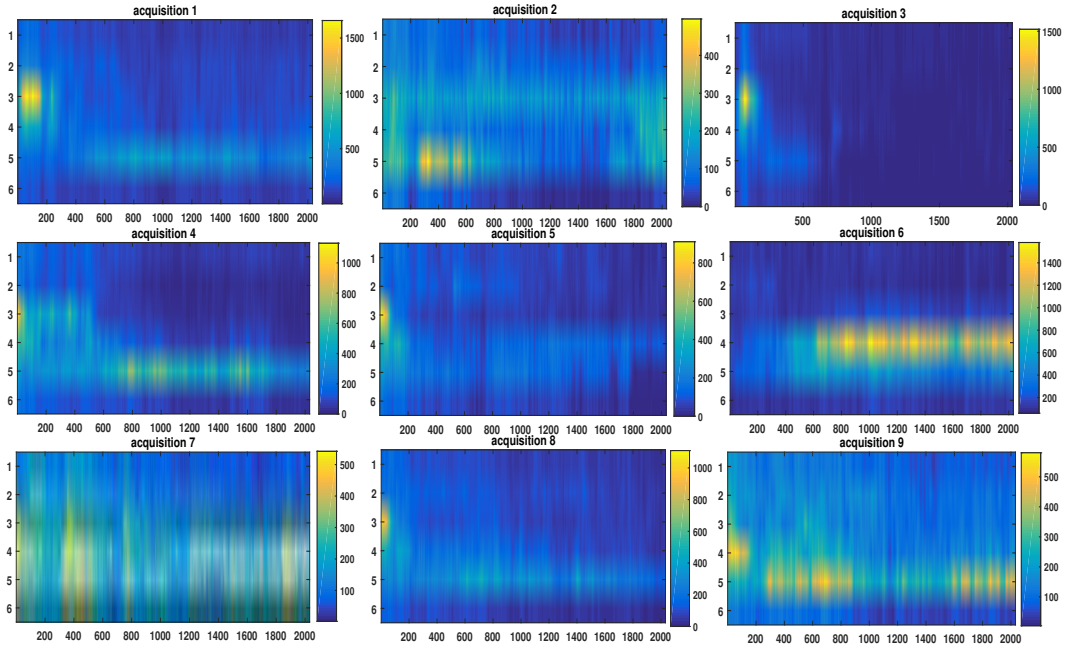

Figure 3.9: EMG signal amplitude of fist hand gesture. The amplitude values are represented in color scale, with respect to the time istances (1-2000) and to the six electrodes

To sum up, we describe the fundamental steps of the new classification method in Table (3.3), with the resulting classification accuracy for each subject reported in Table (3.4).
In particular, the accuracy rate is computed on the validation samples as

$$AC = \frac{n\ of\ validation\ data\ correctly\ classified}{total\ n\ of\ validation\ samples} \cdot 100\%$$

The sensor-features classification method

Given the dataset of an entire registration of a single patient, maintaining the electrodes contributions separated, order the signals according to the gestures' order.

1. Compute the time-features previously defined:
   - x^N normalized signal
   - x' derivative of the signal
   - En energy
   - M local mean
   - V local standard deviation
2. Compute for each pair of acquisitions i, j, the following distances:
   - $d_1 = \|x^N(i) - x^N(j)\|_2^2$
   - $\overline{d} = \|x'(i) - x'(j)\|_2^2 \rightarrow d_2 = exp(-(\overline{d}^4)^{0.00001})$
   - $d_3 = (En(i) - En(j))^2$ or $d_3 = |En(i) - En(j)|$
   - $d_4 = \|M(i) - M(j)\|_2^2$
   - $d_5 = \|V(i) - V(j)\|_2^2$
3. Construct a global array of such distances, of dimension $(nc \cdot np) \times (nc \cdot np) \times ns$, where nc is the number of repetitions for each gesture, np is the number of gestures, ns is the number of electrodes.
4. Divide the total of $nc \cdot np$ acquisitions in training and validation set:
   Training set $\rightarrow$ consider the first 60% of data recorded from each gesture
   Validation set $\rightarrow$ consider the remaining 40% of data
5. Choose the best configuration of 4 pair features-distances, in order to obtain an optimal separation of the diagonal blocks from the distance matrices
6. Validate the features selected on the validation set and compute the accuracy rate of classification, comparing with the desired target labels

Table 3.3: Summary of the classification method

Subject	Classification accuracy
S1	60%
S2	95%
S3	65%
S4	100%
S5	95%
S6	65%
S7	75%
S8	75%
S9	45%
S10	85%

Subject	Classification accuracy
S11	80%
S12	40%
S13	65%
S14	100%
S15	65%
S16	65%
S17	75%
S18	80%
S19	25%
S20	80%

Table 3.4: Classification accuracy from the method of Table (3.3) applied on the sample of 20 trans-radial amputee subjects

In Table (3.4) may be found the numerical results obtained from the method described in Table (3.3), on the totality of the 20 trans-radial amputee subjects. From these results, some observations can be made. Firstly, we note that 10% of the samples reveal a perfect classification, namely S4 and S14, while 5% of the samples performs bad results with an accuracy under 40%. Except for two patients, the remnants have a rate of classification between 60% and 95%. As a preliminary step, these results seem to be promising and may be increased. We proceed in two ways: one is the analysis of threshold signals, with respect to the rest gesture. The other consists of focusing on particular subjects analyzing their results.

In a first time our interest is to compare the signals with respect to the rest gesture level. The motivation is that, in experiment session, the rest plays the same role of the zero-level, and data below the threshold may be considered as noise. We want thus to individuate the presence of acquisitions below the threshold, because they could influence in a negative way our classification. This assumption is supported by a first visul inspection on the signals as they are recorded, comparing the five gestures for each subject. In fact, the EMG signal is the eletric manifestation of a neuromuscolar activation and, in physical terms, it may be considered as the necessary energy for moving the fingers or the art involved in the required gesture. This also justify the reason why we compute the energy as threshold value, as shown below.

A summary of the rest-threshold procedure is reported in Table (3.5), while its application on the entire dataset reveals the results reported in Table (3.6).

The rest-threshold procedure

Given k acquisitions of the rest gesture, compute the threshold value M as the maximum value of the energy associated to these acquisitions.

For $j = 1, ..., nr$ acquisitions of any other gesture, compute
- N the maximum value of the energy of acquisition j
- if $N > \frac{M}{2}$
  - acquisition j above threshold $\rightarrow$ the patient is executing a gesture above the zero-level
- else
  - acquisition j below threshold $\rightarrow$ remove j from the dataset

Table 3.5: Summary of the rest-threshold procedure for trans-radial amputee subjects

Subject	N of acquisitions subthreshold	Old Accuracy
S1	6	60%
S5	1	95%
S12	1	40%
S18	7	80%
S19	5	25%

Table 3.6: Numerical results of the rest-threshold procedure described in Table (3.5), with rereference to the corresponding accuracy rate of Table (3.4)

Once normalized the signals with respect to the rest gesture, what we expect is an increase in the accuracy rate. Firstly, we compare the acquisitions detected above threshold matching the EMG signals plot, and in affirmative case we do a new classification. We now focus on every subject that appears in Table (3.6), describing each particular situation.

Subject 1

We have observed that applying the procedure of threshold in Table (3.5), six acquisitions reveal to be below the threshold, but they do not have a graphic confirmation from the amplitudes of the EMG signals. Desiring to justify this result, we may focus on the amplitudes of the rest EMG signals as shown in Fig. (3.10) and we note that within the same gesture, the amplitude of the signals vary in different ranges. This is not we expect to see, since within the same gesture, the goal is to repeat the gesture as similar as possible, with the same space orientation, sensor location and strength activation.
For this specific subject, it seems thus to be more suitable to consider the mean value of the energy rather than the maximum, as threshold value. In fact, with this assumption, we obtain an average of the activation level of the rest gesture repetitions, which takes into account all the different levels of activation. We could say that the first, third, fourth and sixth acquisitions are not comparable with the others, and we thus may consider them as wrong acquisitions inside the rest gesture. In our opinion, this assumption is too restrictive, also because we would remove 40% of acquisitions without any other hypothesis on the patient.
In conclusion, we decide to treat this patient differently from the others, substituing the threshold condition of Table (3.5) as follows: we consider M equals to the mean value of the energy of the rest acquisitions, and N equals to the mean value of the energy of each other acquisition. Therefore, the classification accuracy is still equal to 60%.

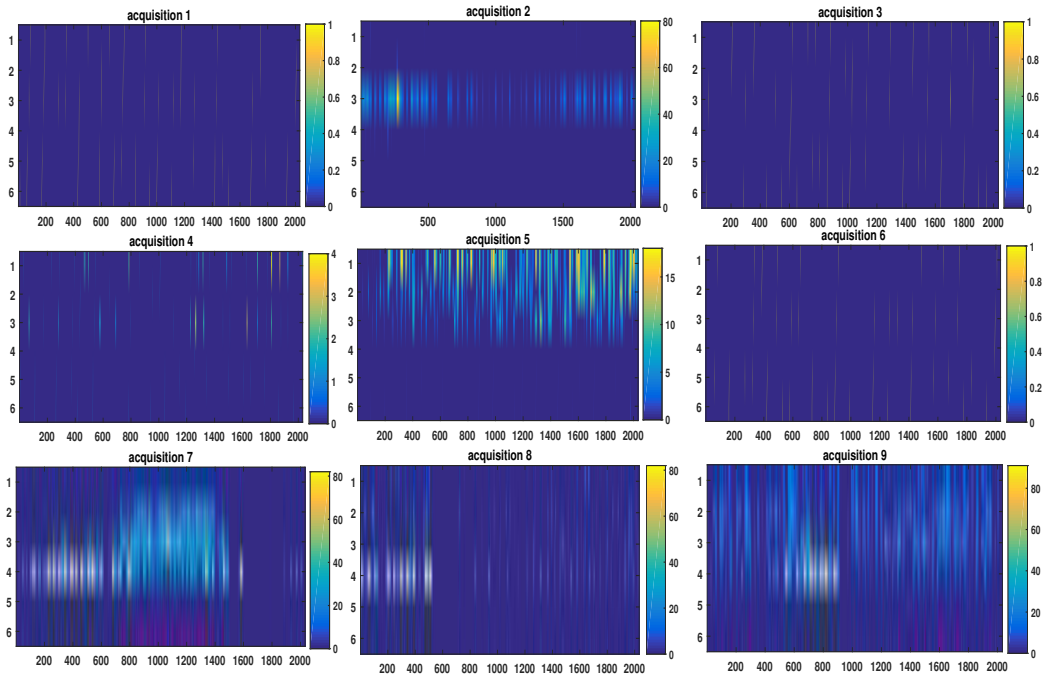

Figure 3.10: Subject 1, nine consecutive repetitions of the rest gesture

Subject 5

In this case, the removal of one single acquisition from the entire dataset would involve a decrease of 30% in the accuracy rate. This contradicts our expectations, since removing bad data should improve the classification accuracy. Therefore, we focus on the acquisitions of the rest gesture, as shown in Fig. (3.11).
In this case, we find the third and fifth acquisitions completely different from the remnants, and we may suppose that these are wrong repetitions of the actual hand gesture. In fact, removing these from the rest gesture, we have no registrations below the threshold, so and the classification accuracy remains equal to 95%.

Subject 12

He is the only patient for whom the removal of one single acquisition involves an increase of the classification accuracy, precisely from 40% to 60%.

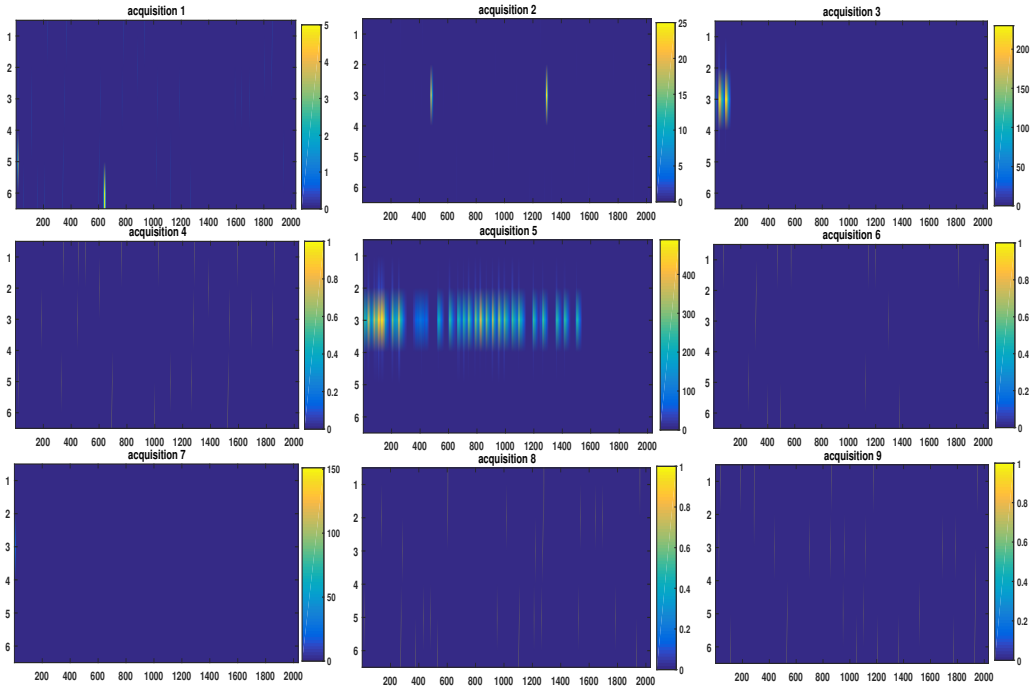

Figure 3.11: Subject 5, nine consecutive repetitions of the rest gesture

Subject 18

In this case it seems that the patient is not able to control his arm. This is evident if we look at the rest gesture, depicted in Fig. (3.12) where no acquisition is equal to the other, and he has strong and brief contractions during this gesture. Moreover, he reveals strong irregularities in executing all other desired hand gestures. Thus, consider the maximum value of the energy of the rest should be not significant for the purpose of our analysis, since it could a be a high value at which the patient is not actually at rest.
As a result, we consider to not include this patient in our analysis, although his preliminary accuracy was equal to 80%.

Subject 19

We consider this patient a confusing patient, who has great difficulty in executing the different hand gestures. In fact, we do not observe any diagonal blocks and sharp distance matrices. We report in Fig. (3.13) and (3.14) the configuration of the four best configuration of features, before and after the removal of the subthreshold acquisitions.

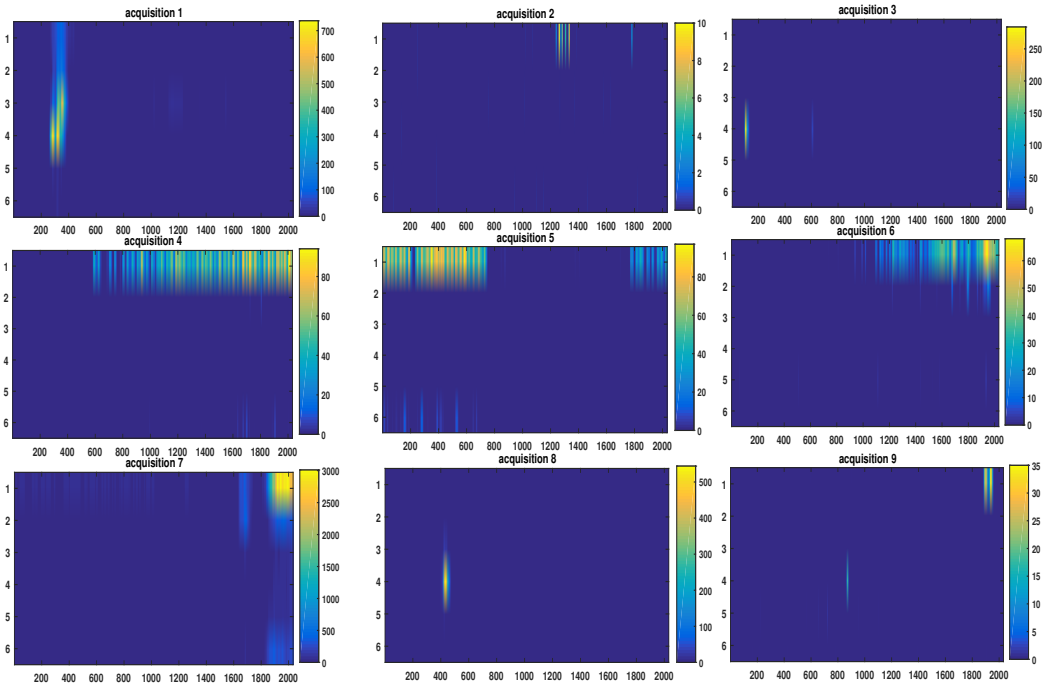

Figure 3.12: Subject 18, nine consecutive repetitions of the rest gesture

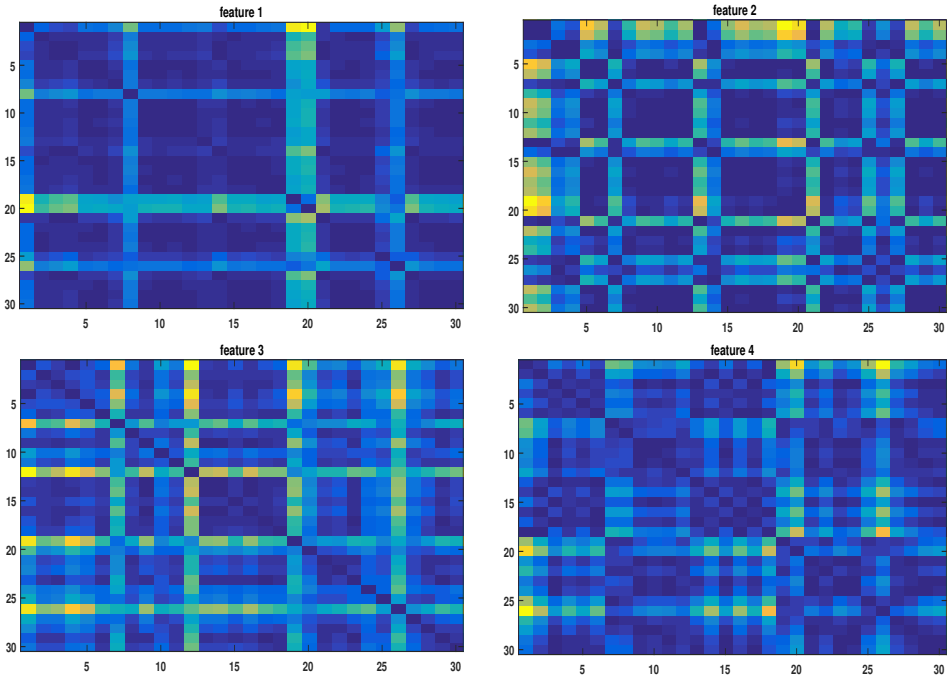

Figure 3.13: Subject 19, best features configuration on the training set composed by the first 60% of registrations

Although we observe an accuracy increase of 5% after removing the acquisitions below the threshold, we decide to not include this patient in our study,

considering him not suitable for pattern recognition prosthesis control.
In conclusion, we summarize the results from Table (3.4) and (3.5) with the previous observations in Table (3.7).
The overall accuracy is therefore computed as

$$Ac = \frac{\sum_{i=1}^{N} Class.\ accuracy}{N} = 75\%$$

with $N = 18$, since we neglected Subject 18 and 19 from our analysis.

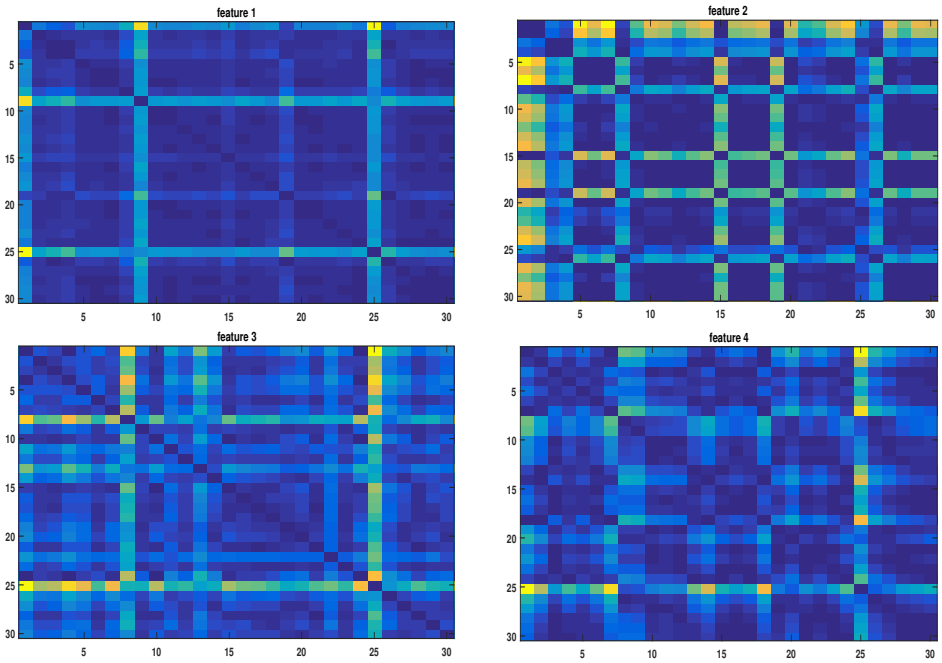

Figure 3.14: Subject 19, best features configuration on the training set composed by the first 60% of registrations, after threshold procedure

Subject	Ac
S1	60%
S2	95%
S3	65%
S4	100%
S5	95%
S6	65%

Subject	Ac
S7	75%
S8	75%
S9	45%
S10	85%
S11	80%
S12	60%

Subject	Ac
S13	65%
S14	100%
S15	65%
S16	65%
S17	75%
S20	80%

Table 3.7: Classification accuracy of the corrected-classification method applied on the accepted 18 trans-radial amputee subjects

3.3 Conclusions and future developments

In this last section we explain some considerations based on our analysis on able-bodied and amputee subjects. We first suggest some motivations behind the different results obtained in the Fourier space, then we underly the importance of performing a good acquisition procedure, proposing some adjustments. At the end, we suggest a way for improving the feature selection as a minimization problem, for which we have obtained preliminarly good results.

3.3.1 EMG signals of amputees in the Fourier space

We have observed that analyze the EMG amputee signals in the Fourier space does not reveal any kind of pattern classification. This result was somewhat expected. In fact, in order to study the frequency behavior of the signals, it should probably be more useful to consider the signal in its raw state. This is motivated by the fact that the surface electrodes used in this work just execute an amplification and sampling pre-processing. Thus, because of the frequency limit, they elaborate the signal and pass it amplified and sampled to the PC. In this context we deal with a contentious issue: on the one hand, for analyzing EMG signals from the frequency domain point of view, it should be necessary to have rough signals; on the other hand, dealing with signals at their raw state is not useful from the application point of view.

However, we propose a way in which the Fourier analysis could reveal interesting results. It has been proved that EMG signals from amputee subjects improve with the experience [15]. This means that, giving a myolectric prosthesis to a patient for a training phase, we expect to have better signals after a training step. Moreover, the constraint of prosthesis real-time execution plays a central role in signals improvement. While the tridigit prosthesis response time is equal to 40 ms, the myoelectric prosthesis response time is expected to be within 100 ms, and when the movement starts to be performed, the patient will undoubtly make corrections and change the impulse if the movement performed by the device does not correspond to the one desired. Thus, when better signals are recorded, we would expect that Fourier analysis provides better results rather than what we exposed in this chapter, in a similar way as for able-bodied subjects.

Since we have noted an evident difference between healthy subjects and transradial amputee signals, we may focus on different aspects which could be the cause

of the hard upper limb control. For instance, there may be a central nervous system problem, which supervises the main control and processing functions, that could cause a not clear mapping of the intentions. Otherwise, it could be a surgical problem, depending on the way the muscles and nerves were operated and thus the possibility of their damage and reinnervation. The justification of this deep difference is still an argument of investigation.

In any case, we have to underline that there is the possibilty for some patients to be not suitable for pattern recognition based system control. With reference to this observation, in Section 2.2 we have decided to neglect Subject 18 and Subject 19 from our study.

3.3.2 Acquisition procedure and individuation of pattern recognition suitable patients

In our opinion, the acquisition procedure is a complicated and not uniquely well defined step. We have observed that it is common to detect transient signals at the external phases of some recordings, which are detectable as more intense band rather than the remnant registration. In order to execute a good acquisition, the operator should follow some steps as described below. In the acquisition phase, we have used a monitor interface which depicted the EMG plots for each sensors. This support helps the operator to detect when the patient is executing the asked gesture. In simple terms, the operator should ask the patient to perform the gesture, analyzing the activation of the sensors and waiting for few seconds in order to let the signal to regularize and let the transient to pass. Then, the gesture is acquired for the acquisition time decided preliminarly (in our study, we have always considered an acquisition time equals to 2 seconds) and do the same for all the acquisition phase. In this way, we would have as much as possible steady-state signals, from which we could normalize the entire dataset with respect to the zero-level of the acquisition procedure (in our analysis, we have standardized the signals with respect to the rest gesture).
Thanks to this normalization procedure, we have a strategy for the individuation of pattern recognition suitable patients. In fact, in the acquisition procedure we are able to compute the threshold level of activity, as described by the procedure of Table (3.5), on the basis of the rest gesture acquisition. Each following acquisition can be compared to this zero-level value, allowing the operator to individuate effective gestures and below threshold ones, which can be considered as noise

component. At the time this procedure ends, we have the number of wrong data with respect to the rest gesture, which can be confirmed by a visual matching with the amplitudes of the EMG signals. In this terms, it is operator's choice to make corrections on the rest-threshold procedure: the threshold condition can be modified by considering the mean value of the energy rather than the maximum value, as we have done in our study for Subject 1, or abnormal rest acquisitions can be removed, if they totally disagree from the others, or more simply he can decide to neglect the subject since he is not able to make dinstict signals, considering him a non suitable pattern recognition patient.

3.3.3 Improvements of the classification

From the application point of view, we suggest three possible ways to increase our classification procedure. One concerns the data segmentation in time windows in order to work with a major number of data, the other is about a different procedure in data partition and finally an automated scheme for the optimal feature extraction.

Windowing technique

As a first observation, working with time windows of 1/10 of elements rather than a complete acquisition could represent an improvement from the point of view of accuracy rate. This approach would have a double benefit: on the one hand, it allows us to work with a major number of data, for each of which we compute the features, on the other hand it has a more real-time applicability, since we should not wait for the end of a recording bur we calculate the features a little at time.

Data partition

A second observation concerns a different partitioning procedure for training and test subsets. The results presented in Section 2.2 were based on an arbitrary selection of training and validation data: in fact, we considered, for each gesture, the first six acquisitions as training set and the remaining four acquisitions as validation set. In order to let the classifier to achieve robustness and generalization properties, we should split the dataset in a shuffle modality, always considering the percentages of 60% as training and 40% as validation, but in all the possible combinations within the totality of registrations.

Automatic selection of the best sensor-feature pairs

We describe now a way to improve the features extraction procedure, based on an automatic search of them, not only based on the visual inspection and detection but formulated as a minimization problem.
We want to obtain squared distance matrices whose diagonal marked blocks correspond to the hand gestures involved in the experiment, choosing the best configuration of distance matrices and electrodes. Until now we have chosen manually the features which best separated each diagonal block from the others extra-diagonal. In order to improve the method and its robustness, we may implement an features automatic search on the training set as a minimization problem. For a better clarification, our goal is to obtain a matrix as depicted in Fig. (3.15). In this context, we would obtain blocks on the principal diagonal which values are closest to zero. They represent the distances computed between each pair of acquisitions of the same gesture, so being closed to zero, they may be identified and are able to individuate each gesture. The extra-diagonal blocks would be instead far from zero, representing the distances between pairs of different gestures.

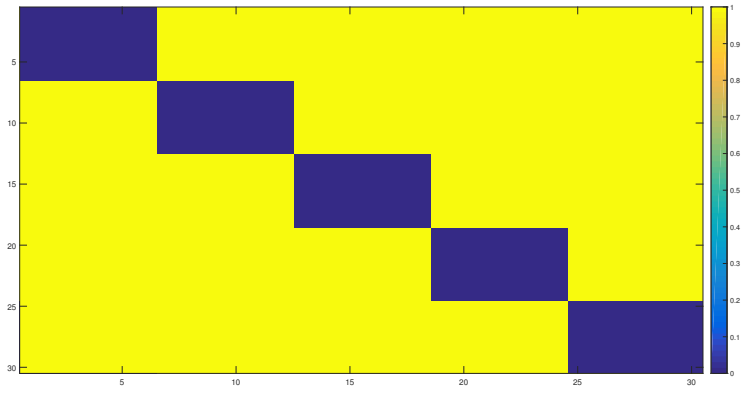

Figure 3.15: Target diagonal block matrix

In our study, we have considered five temporal features for each of the six surface electrodes involved in the acquisition phase. Therefore, we have to search for the best combination of features within 30 possible features.
Let denote each feature-distance matrix as H_i, with $i = 1, ..., 30$, and for each one let introduce a weight coefficient denoted as h_i, with $i = 1, ..., 30$.
Let introduce the target distance block matrix referring to the training set as a

matrix Σ of 30×30 elements, as

$$\Sigma = \begin{pmatrix} 0 & s_1 & s_2 & s_3 & s_4 \\ s_1 & 0 & s_5 & s_6 & s_7 \\ s_2 & s_5 & 0 & s_8 & s_9 \\ s_3 & s_6 & s_8 & 0 & s_{10} \\ s_4 & s_7 & s_9 & s_{10} & 0 \end{pmatrix}$$

where each matrix element s_i, $i = 1, ..., 10$ corresponds to a block of six elements. We want to minimize the functional cost defined by

$$J = \|h_1 \cdot H_1 + h_2 \cdot H_2 + ... + h_{30} \cdot H_{30} - \Sigma\|_2^2 = \| \sum_{i=1}^{30} h_i \cdot H_i - \Sigma\|_2^2 \qquad (3.2)$$

In other words, we search the best combination of distance matrices so that the squared Euclidean norm between their linear combination and the target Σ is minimal.
We implement this method using the predefined Matlab function `fmincon` (https://it.mathworks.com/help/optim/ug/fmincon.html) for constrained optimization problems, , given an initial point h_0 for the weight coefficients and the extra diagonal blocks all equal to the constant value 1.

At last, we suggest from the one hand to not incorporate in the functional cost defined by Eq. (3.2) the matrices which do not contain any kind of information, that are the ones which are not able to detect blocks corresponding to the gestures and maybe the most responsible of noise content. On the other hand, we could also start with the ones able to individuate some marked blocks, and add the others at little at time, looking for improvements in the optimal solutions.

Bibliography

[1] Benatti S., Farella E., Gruppioni E., Benini L., *Analysis of Robust Implementation of an EMG Pattern Recognition based Control*, BIOSIGNALS (2014)

[2] Bishop M. C., *Neural networks for pattern recognition*, Oxford University Press (1995)

[3] Bishop M. C., *Pattern recognition and machine learning*, Springer (2006)

[4] Castellini C., Gruppioni E., Davalli A., Sandini G., *Fine detection of grasp force and posture by amputees via surface electromyography*, Journal of Physiology-Paris 103.3 (2009): 255-262

[5] De Luca C., *Electromyography*, Encyclopedia of Medical Devices and Instrumentation, (2006): 98-109

[6] Englehart K., Hudgins B., Parker P.A., *A wavelet-based continuous classification scheme for multifunction myoelectric control*, IEEE Transactions on Biomedical Engineering 48.3 (2001): 302-311

[7] Englehart K., Hudgins B., Parker P.A., Stevenson M., *Classification of the myoelectric signal using time-frequency based representations*, Medical engineering & physics 21.6 (1999): 431-438

[8] Guanglin L., *Electromyography pattern-recognition-based control of powered multifunctional upper-limb prostheses*, INTECH Open Access Publisher, 2011

[9] Haykin S., *Neural Networks and Learning Machines, 3rd Ed.*, Pearson (2008)

[10] Hiraiwa A., Uchida N., Shimohara K., *EMG pattern recognition by neural networks for prosthetic fingers control*, Annual Review in Automatic Programming 17 (1992): 73-79

[11] Hyvärinen A., Karhunen J., Oja E., *Indipendent Component Analysis*, John Wiley & Sons, (2001): 125-138

[12] Meila M., Shi J., *A random walks view of spectral segmentation*, 8th International workshop on artificial intelligence and statistics (AISTATS) (2001)

[13] Oskoei M. A., Huosheng H., *Myoelectric control systems—A survey*, Biomedical Signal Processing and Control 2.4 (2007): 275-294

[14] Reaz M. B. I., Hussain M. S., Mohd-Yasin F., *Techniques of EMG signal analysis: detection, processing, classification and applications*, Biological procedures online 8.1 (2006): 11-35

[15] Riillo F., Quitadamo L.R., Cavrini F., Saggio G., Sbernini L., Pinto C.A., Pastò N.C., Gruppioni E., *Evaluating the influence of subject-related variables on EMG-based hand gesture classification*, Medical Measurements and Applications (MeMeA), 2014 IEEE International Symposium on IEEE (2014)

[16] Riillo F., Quitadamo L.R., Cavrini F., Gruppioni E., Pinto C.A., Pastò N.C., Sbernini L., Albero L., Saggio G., *Optimization of EMG-based hand gesture recognition: Supervised vs. unsupervised data preprocessing on healthy subjects and transradial amputees*, Biomedical Signal Processing and Control 14 (2014): 117-125

[17] Rossi M., Benatti S., Farella E., Benini L., *Hybrid EMG classifier based on HMM and SVM for hand gesture recognition in prosthetics*, Industrial Technology (ICIT), IEEE International Conference on. IEEE, (2015)

[18] Scheme E., Englehart K., *Electromyogram pattern recognition for control of powered upper-limb prostheses: State of the art and challanges for clinical use*, Journal of Rehabilitation Research and Development 48.6, (2011): 643-660

[19] Shi J., Malik J., *Normalized cuts and image segmentation*, IEEE Transactions on pattern analysis and machine intelligence 22.8 (2000): 888-905

[20] Steinwart I., Christmann A., *Support Vector Machines*, Springer (2008)

[21] Tassinary L. G., Caccioppo J. T., Vanman E., *The Skelemotor System: Surface Electromyography*, 267-299

[22] Verni G., Cutti A. G., Gruppioni E., Amoresano A., *Nuove tecnologie e innovazione nelle protesi di arto superiore*, Medicina e chirurgia ortopedica n.3, (2012): 20-33

[23] Von Luxburg U., *A tutorial on spectral clustering*, Statistics and computing 17.4 (2007): 395-416