

UNIVERSITÁ DEGLI STUDI DI PADOVA

CORSO DI LAUREA IN INGEGNERIA INFORMATICA

TESI DI LAUREA MAGISTRALE

---

**Analisi indipendente dal soggetto  
di segnali sEMG per il controllo  
multi-giunto di una mano  
robotica**

---

*Laureando:*  
Daniele BOVO

*Relatore:*  
Emanuele MENEGATTI

*Correlatore:*  
Stefano MICHIELETTO

10 Aprile 2017





# Abstract

In the last years has increased the number of papers about analysis and development of mechanical and robotic prosthesis driven by mean of biometric signals. One of the most followed approaches is to use muscle signals, namely surface electromyography (sEMG) signals, to generate probabilistic models that, once trained, will be able to predict the limb motion starting from new sEMG signals never used before, to train the model. In such a circumstance, the goodness of the estimation and the efficiency of the model depend on how the most significant channels (i.e. electrodes) are chosen.

This thesis follows step by step the development and analysis of an algorithm that aims to select most significant sEMG channels in order to use them as input to a Gaussian Mixture Model (GMM). This probabilistic model consists of a weighted sum of gaussian components. It will be fed with sEMG signals coming from channels that the outlier rejection algorithm will return as output and with joints signals involved in the considered movement. The goal of this work is to replace the qualitative, manual, best channels choice with an automatic choice made by an algorithm.

The whole project uses multiple joints signals coming from multiple subject's hand. The goodness of the algorithm is indirectly computed by mean of accuracy of predictions made on the base of the GMM model. Building the model upon multiple subjects signals allows us to generate a robust, subject-independent and general structure. Moreover, movements will be estimated with a regression technique called Gaussian Mixture Regression (GMR) starting from EMG signals not used for the model training. Subject-independent property allows the system to be used by novel subjects without any initial training phase.

The algorithm is divided in four parts: data extraction and pre-processing,

outliers detection, results computation and accuracy computation. Once the algorithm core has been defined, we have investigate different improvements to each phase in order to refine it. The code has been written in Matlab, C++ and minimally R language.

# Sommario

L'analisi e lo sviluppo di protesi robotiche e meccaniche controllate per mezzo di segnali biometrici sta diventando un argomento di studio sempre più trattato. Uno degli approcci più seguiti è quello di utilizzare i segnali provenienti dai muscoli, nella fattispecie segnali elettromiografici superficiali (sEMG), per generare modelli probabilistici che, una volta allenati, riusciranno a prevedere il movimento dell'arto a fronte di un nuovo segnale sEMG non utilizzato per allenare il modello. In un contesto simile la scelta dei canali (i.e. elettrodi) più informativi che compongono il sistema sEMG ricopre un ruolo essenziale: da questa decisione dipenderanno l'efficienza del modello e la bontà della stima del movimento che questo produrrà.

In questo elaborato vengono seguiti passo dopo passo lo sviluppo e l'analisi di un algoritmo per la selezione dei migliori canali sEMG da fornire in ingresso ad un modello probabilistico Gaussian Mixture Model (GMM). Esso è composto da una somma pesata di componenti gaussiane e verrà allenato con i segnali relativi ai canali del sottoinsieme ottimale specificato dall'algoritmo e con gli angoli assunti nel tempo dai giunti coinvolti nel movimento. Finora la scelta dei canali migliori veniva fatta in maniera qualitativa e poco formale, l'algoritmo proposto in questa tesi ha lo scopo di sollevare l'utente dal compito di scegliere manualmente una combinazione di canali ideale per la generazione del modello GMM.

L'intero progetto utilizza segnali relativi a movimenti multi-giunto (che prevedono cioè l'utilizzo di più articolazioni) della mano di diverse persone. La bontà dei canali selezionati dall'algoritmo è misurata indirettamente tramite l'accuratezza delle predizioni fatte sulla base del modello GMM. Costruendo il modello GMM su dati di soggetti diversi generiamo una struttura sufficientemente robusta e generica, indipendente dal soggetto. A conferma di ciò, i movimenti saranno stimati con una tecnica di regression (GMR) partendo da segnali EMG di soggetti non utilizzati per allenare il modello. L'indipendenza dal soggetto fa sì che il siste-

ma possa essere utilizzato da soggetti nuovi senza bisogno di sottoporli a lunghe sessioni di training.

Dal punto di vista operativo, l'algoritmo è stato suddiviso in quattro fasi principali: estrazione dei dati e pre-processing, outlier detection, elaborazione dei risultati ed infine calcolo dell'accuratezza delle predizioni. A posteriori, dopo aver definito una traccia per il funzionamento dell'algoritmo, sono state investigate diverse modifiche alle varie fasi con lo scopo di migliorare i risultati ottenuti con la prima versione, più grezza, dell'algoritmo. Il codice utilizzato è stato sviluppato usando i linguaggi Matlab, C++ ed in minima parte R.

# Acronimi

**CWT** Continuous Wavelet Transform  
**DTW** Dynamic Time Warping  
**DWT** Discrete Wavelet Transform  
**EM** Expectation Maximization  
**EMG** Elettromiografia  
**FMCD** Fast Minimum Covariance Determinant  
**GMM** Gaussian Mixture Model  
**GMR** Gaussian Mixture Regression  
**GOF** Goodness Of Fit  
**MAD** Mean Absolute Deviation  
**MU** unità motoria  
**MCD** Minimum Covariance Determinant  
**Ninapro** Non-Invasive Adaptive Hand Prosthetics  
**NMSE** Normalized Mean Square Error  
**OGK** Orthogonalized Gnanadesikan-Kettenring  
**PCA** Principal Component Analysis  
**sEMG** Elettromiografia Superficiale  
**xcorr** Cross Correlation  
**xcov** Cross Covariance



# Indice

<b>1</b>	<b>Introduzione</b>	<b>11</b>
1.1	Obiettivi e motivazione . . . . .	11
1.2	Contesto applicativo . . . . .	13
1.3	Gaussian Mixture Model . . . . .	13
1.3.1	Gaussian Mixture Regression . . . . .	15
1.4	Elettromiografia e sEMG . . . . .	16
<b>2</b>	<b>Dataset</b>	<b>19</b>
2.1	Descrizione . . . . .	19
2.2	Data Set . . . . .	20
2.3	Estrazione dei dati . . . . .	22
<b>3</b>	<b>Stato dell'arte</b>	<b>25</b>
3.1	Selezione di canali in un sistema sEMG . . . . .	25
3.2	Modelli probabilistici per il movimento di protesi robotiche a partire da segnali biometrici . . . . .	27
<b>4</b>	<b>Pre-processing</b>	<b>29</b>
4.1	Trasformata Wavelet . . . . .	29
4.2	Elaborazione dei segnali . . . . .	32
<b>5</b>	<b>Outliers detection</b>	<b>35</b>
5.1	Hampel filter . . . . .	35
5.2	Gaussian distribution fitting . . . . .	38
5.3	Outliers detection basata su PCA . . . . .	39
5.3.1	Principal Component Analysis (PCA) . . . . .	39
5.3.2	Robust multivariate covariance estimate . . . . .	41
5.3.3	Bivariate gaussian distribution fitting . . . . .	44

<b>6</b>	<b>Processo di selezione</b>	<b>47</b>
6.1	Scelta preliminare dei canali . . . . .	47
6.2	Somiglianza tra segnali: stato dell'arte . . . . .	50
6.3	Cross-covariance nell'analisi dei segnali . . . . .	52
6.4	Metodo di selezione sistematico . . . . .	53
6.4.1	Algoritmo di selezione . . . . .	55
6.4.2	Interpretazione e commento dei risultati . . . . .	57
6.5	Verifica della correttezza . . . . .	57
<b>7</b>	<b>Analisi dell'algoritmo</b>	<b>61</b>
7.1	Combinazione delle elaborazioni . . . . .	61
7.2	Metodo di valutazione pesato . . . . .	63
7.3	Normalizzazione sul canale . . . . .	65
7.4	Allineamento dei segnali . . . . .	70
7.5	Prestazioni . . . . .	72
<b>8</b>	<b>Risultati</b>	<b>73</b>
8.1	Allenamento del modello GMM e regressione . . . . .	73
8.2	Misure di accuratezza . . . . .	74
8.3	Analisi dei risultati . . . . .	76
<b>9</b>	<b>Conclusioni</b>	<b>79</b>
9.1	Discussione . . . . .	79
9.2	Sviluppi futuri . . . . .	80

# Capitolo 1

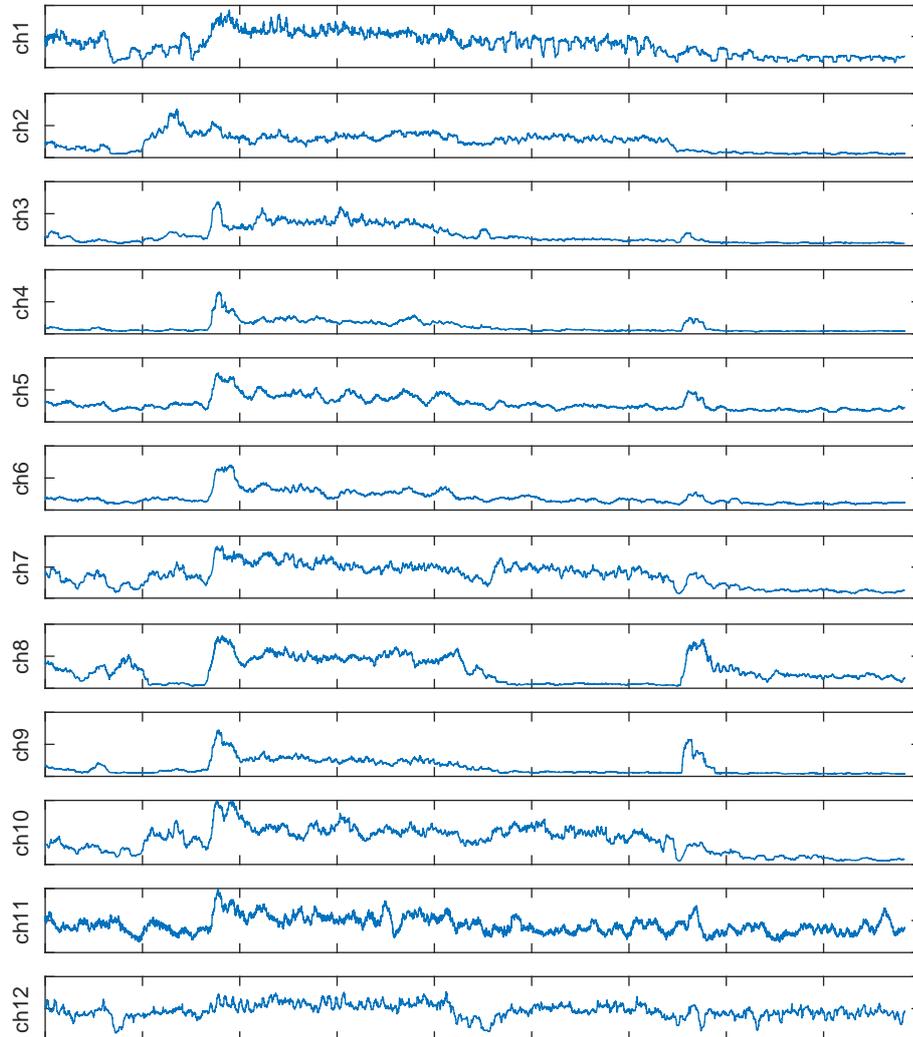
## Introduzione

### 1.1 Obiettivi e motivazione

Lo scopo di questo elaborato è quello di illustrare il processo di selezione dei canali elettromiografici più promettenti tra tutti quelli che registrati tramite un sistema di scansione elettromiografica superficiale (sEMG) al fine di costruire un modello di movimento probabilistico basato su Gaussian Mixture Model (GMM) per il comando di una protesi robotica tramite i segnali sEMG stessi. L'obiettivo principale è quello di individuare un metodo sistematico per la selezione dei canali più significativi che sostituisca l'osservazione empirica. Avendo a che fare con un dataset composto da una grande mole di dati, risulta preferibile, se non necessario, sviluppare un metodo che assicuri la correttezza della scelta con l'obiettivo di ridurre i tempi di elaborazione.

I segnali EMG sono solitamente composti da più canali, corrispondenti all'attivazione di diversi gruppi muscolari (vedi *sezione 1.4*). L'intrinseca natura multi-canale dei segnali EMG ha sollevato alcune difficoltà riguardo la riduzione del numero di canali sui quali applicare algoritmi di analisi e classificazione del movimento e il problema dell'elevata dimensionalità di questi sistemi, se non trattato, può facilmente portare a cattive prestazioni, sia in termini computazionali che di accuratezza del risultato. Se si pensa poi ai contesti reali di applicazione, come ad esempio al movimento di diversi giunti in una mano robotica, diventa consigliabile, se non necessario, affrontare la questione per ottenere risultati significativi. Scegliere accuratamente i canali in maniera preventiva e processare adeguatamente l'informazione grezza iniziale permette di ottenere buoni risultati

dal modello risparmiando sui tempi di computazione.



In alcuni casi succede che i segnali grezzi che compongono il dataset siano incoerenti con la maggior parte degli altri segnali, e mostrano andamenti molto diversi da quelli che ci si aspetterebbe. Queste differenze, non trascurabili, probabilmente dovute a stanchezza o distrazioni del soggetto esecutore, costituiscono una fonte di errore per il calcolo del modello finale. Questo problema crea la necessità di predisporre una fase di eliminazione di questi segnali "spuri" che viene detta *outlier rejection*.

Per assicurare una certa generalità dell'applicazione si è scelto di mantenere l'indipendenza del sistema dal soggetto con un approccio del tipo *subject-independent*. In questo modo il risultato ottenuto è robusto e generico, in quanto al suo interno contiene tutta la variabilità portata da molti soggetti diversi. Grazie a questo tipo di approccio i risultati che otteniamo possono venire applicati a soggetti differenti senza bisogno di lunghe fasi di train. Sempre con lo scopo di realizzare un sistema il più robusto e generico possibile, abbiamo cercato anche di mantenerlo indipendente dal movimento considerato (*movement-independent*). Movimenti che coinvolgono giunti diversi nello stesso arto, ma anche in arti differenti (ad esempio mano e ginocchio) vengono sottoposti alla stessa fase di pre-processing e modellizzazione. Questo ci consente di ampliare rapidamente il numero e il tipo di segnali studiati. Tutti i movimenti considerati, inoltre, sono complessi e multi-giunto.

## 1.2 Contesto applicativo

Come specificato nella sezione precedente, l'intero algoritmo è pensato per essere applicato ad un modello per il comando di protesi robotiche tramite segnali di Elettromiografia Superficiale (sEMG). A valle della selezione dei canali è prevista la costruzione di un modello probabilistico GMM, allenato con i segnali sEMG provenienti dai canali selezionati [30][31].

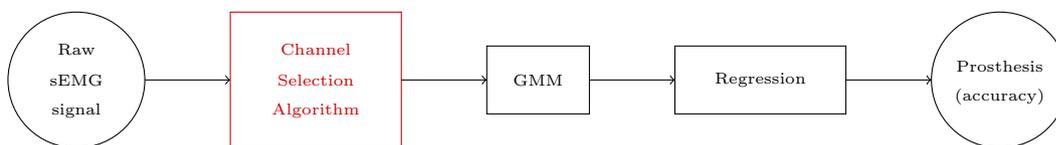


Figura 1.1: Diagramma dell'intero sistema di attivazione di una protesi robotica. Il blocco innovativo introdotto in questa tesi è evidenziato con il colore rosso.

Inserire l'algoritmo all'interno di una catena come quella rappresentata in *Figura 1.1* permette di sfruttare l'accuratezza introdotta a posteriori dalla regressione come ulteriore indice di valutazione per i risultati dell'algoritmo.

## 1.3 Gaussian Mixture Model

L'utilizzo di un modello GMM permette di ottenere la somma pesata di  $K$  componenti gaussiane che approssimino, per un determinato movimento, il dataset

formato da segnali sEMG e segnali relativi ai movimenti dei giunti della mano. Quali saranno i segnali da utilizzare per costruire questo dataset, ce lo dirà l'algoritmo presentato in questa tesi, come risultato della sua elaborazione. Nel nostro caso il dataset usato per allenare il modello GMM, detto *training set*, sarà composto da  $N = nT$  record, dove  $n$  è il numero di trial considerati e  $T$  è il numero dei campioni (i.e. osservazioni) che compongono ogni trial. In simboli:

$$dataset = \{\zeta_j, j = 1, \dots, N\} \quad (1.1)$$

dove il singolo trial  $\zeta_j$  è definito come:

$$\zeta_j = \{\xi(t), \alpha(t)\} \in \mathbb{R}^D \quad (1.2)$$

Come accennato poco fa le due componenti del singolo record  $\xi(t)$  e  $\alpha(t)$  fanno riferimento, per un determinato tempo  $t$ , ad un certo numero  $C$  di valori di sEMG e ad un certo numero  $G$  di giunti della mano ,rispettivamente. In particolare, per i segnali sEMG vale:

$$\xi(t) = \{\xi_1(t), \xi_2(t), \dots, \xi_C(t)\} \in \mathbb{R}^C \quad (1.3)$$

mentre per i giunti della mano vale:

$$\alpha(t) = \{\alpha_1(t), \alpha_2(t), \dots, \alpha_G(t)\} \in \mathbb{R}^G \quad (1.4)$$

da cui si deduce che  $D = C + G$  e che  $\xi_i(t) \in \mathbb{R}, \forall i \in \{1, \dots, C\}$  e, analogamente,  $\alpha_j(t) \in \mathbb{R}, \forall j \in \{1, \dots, G\}$ . Ciò significa che l'output che ci si aspetta è una mistura di  $K$  variabili aleatorie gaussiane multivariate, definite cioè su  $D$  dimensioni.

Il modello GMM è costruito grazie all'algoritmo Expectation Maximization (EM) [40] che ci permette di ottenere una distribuzione di probabilità a partire dal training set, usata successivamente assieme al *validation set* per calcolare la

regressione sui giunti della mano selezionati. L'algoritmo EM stima iterativamente i parametri ottimi  $\theta_i = (\mu_i, \Sigma_i), i \in \{1, \dots, K\}$  che caratterizzeranno le  $K$  variabili aleatorie gaussiane che andranno a formare il modello GMM, dove  $\mu_i$  e  $\Sigma_i$  rappresentano rispettivamente il vettore media e la matrice di covarianza per la  $i$ -esima variabile aleatoria gaussiana multivariata in  $D$  dimensioni. L'algoritmo è suddiviso in due fasi racchiuse in un ciclo: la fase di *Expectation* e la fase di *Maximization*. Tale ciclo si ferma solo quando l'incremento della probabilità logaritmica (log-likelihood)  $\mathcal{L} = \sum_{j=1}^N \log(p(\zeta_j|\theta))$ , che diminuisce ad ogni ciclo, supera la soglia definita dalla disuguaglianza  $\frac{\mathcal{L}(t+1)}{\mathcal{L}(t)} < \epsilon$ .

### 1.3.1 Gaussian Mixture Regression

L'algoritmo Gaussian Mixture Regression (GMR) viene utilizzato per risalire agli angoli di piegamento dei giunti a partire dai segnali sEMG utilizzati come validation set. L'informazione necessaria per concretizzare il processo è contenuta nel modello GMM, una volta allenato con il training-set. Più formalmente, la  $k$ -esima componente della mistura gaussiana è definita dai parametri  $(\pi_k, \mu_k, \Sigma_k)$  dove:

$$\mu_k = \{\mu_{p,k}, \mu_{\alpha,k}\} \quad \Sigma_k = \begin{bmatrix} \Sigma_{p,k} & \Sigma_{p\alpha,k} \\ \Sigma_{\alpha p,k} & \Sigma_{\alpha,k} \end{bmatrix} \quad (1.5)$$

con  $\mu_p$  e  $\Sigma_p$  sono rispettivamente la media e la matrice di covarianza dell'informazione nota a priori. L'angolo di piegamento dei giunti e la relativa covarianza viene stimata rispettivamente con le seguenti equazioni:

$$\hat{\alpha} = E[\alpha|t, \xi] = \sum_{k=1}^K \beta_k \hat{\alpha}_k \quad (1.6)$$

$$\hat{\Sigma}_s = Cov[\alpha|t, \xi] = \sum_{k=1}^K \beta_k^2 \hat{\Sigma}_{\alpha,k} \quad (1.7)$$

dove:

- $\beta_k = \frac{\pi_k \mathcal{N}(t, \xi_c | \mu_{p,k}, \Sigma_{p,k})}{\sum_{j=1}^K \pi_j \mathcal{N}(t, \xi_c | \mu_{p,j}, \Sigma_{p,j})}$  è il peso della  $k$ -esima componente della mistura gaussiana.
- $\hat{\alpha}_k = E[\alpha_k|t, \xi] = \mu_{\alpha,k} + \Sigma_{\alpha p,k} (\Sigma_{p,k})^{-1} (\{t, \xi\} - \mu_{p,k})$  è l'aspettazione di  $\alpha_k$  condizionata sapendo  $\{t, \xi\}$ .

- $\hat{\Sigma}_{\alpha,k} = Cov[\alpha_k|t, \xi] = \Sigma_{\alpha,k} + \Sigma_{\alpha p,k}(\Sigma_{p,k})^{-1}\Sigma_{p\alpha,k}$  è la covarianza di  $\alpha_k$  condizionata sapendo  $\{t, \xi\}$ .

Ciò significa che la stima del movimento di un giunto tramite l'algoritmo GMR necessita solamente di conoscere il peso, la media e la covarianza degli elementi che costituiscono la mistura, calcolati secondo l'algoritmo EM.

## 1.4 Elettromiografia e sEMG

L'Elettromiografia (EMG) è una tecnica che consente la registrazione e lo studio dei segnali elettrici generati dai muscoli. La fibra muscolare, detta anche fibrocellula, è l'elemento caratteristico del tessuto muscolare. Di forma allungata e fusiforme, si contrae, accorciandosi, a fronte di uno stimolo nervoso. Le fibre muscolari sono organizzate in gruppi, ciascuno raggiunto dalle ramificazioni terminali di un singolo motoneurone  $\alpha$ . Il motoneurone e le fibre da esso innervate costituiscono l'unità motoria (MU): la più piccola unità funzionale che descrive l'attività muscolare.

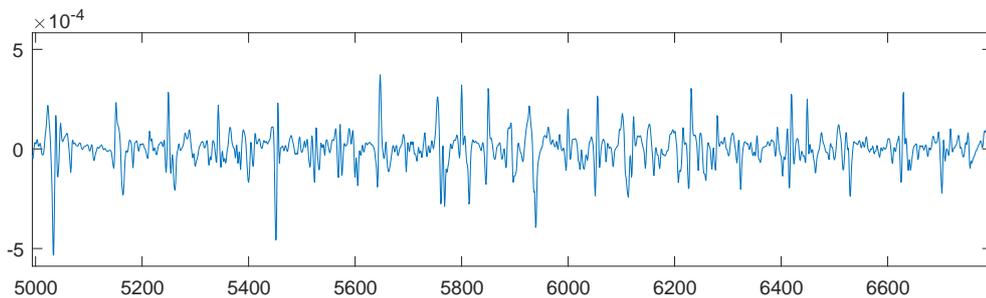


Figura 1.2: Dettaglio di un segnale EMG grezzo.

La contrazione della fibra muscolare è innescata da un impulso elettrico propagato dal motoneurone  $\alpha$ , attivato dal sistema nervoso centrale o da un riflesso, che crea una depolarizzazione delle fibre muscolari le quali, a loro volta, generano dei potenziali elettrici. L'elettromiografia misura proprio questi potenziali, che si formano in un muscolo durante la sua contrazione. I singoli potenziali rispecchiano l'attività di una singola unità motoria nel caso di elettrodi ad inserzione (o ad ago), oppure di un gruppo di unità motorie nel caso di elettrodi di superficie (sEMG).



Figura 1.3: Esempio di applicazione di alcuni elettrodi di un sistema sEMG.

Ogni elettrodo di superficie registra l'attivazione di un singolo muscolo o di un gruppo di muscoli. I segnali registrati da ogni elettrodo corrispondono ai canali da selezionare per essere dati in ingresso al modello di probabilistico GMM e al sistema di regressione GMR.



# Capitolo 2

## Dataset

### 2.1 Descrizione

I dati grezzi che abbiamo scelto di utilizzare per generare il modello GMM sono contenuti nel database Non-Invasive Adaptive Hand Prosthetics (Ninapro)<sup>1</sup> [33][36]: un progetto in continua crescita che ha come obiettivo quello di rendere disponibile gratuitamente alla comunità scientifica un dataset contenente segnali mioelettrici e cinematici della mano e del polso. La comunità scientifica infatti risente della mancanza di dataset ricchi di soggetti e di movimenti su cui fare esperimenti e su cui confrontarsi [34][35]. Ninapro conta attualmente 3 database; nell'ambito di questo elaborato è stato utilizzato il secondo di questi.

L'esperimento descritto dal secondo database è diviso in tre gruppi di esercizi:

1. Movimenti basilari delle dita e del polso
2. Prese e altri movimenti funzionali
3. Schemi di misurazione della forza per le dita

Nei i primi due esercizi l'informazione cinematica è registrata con l'utilizzo di un accelerometro posizionato sul polso e di un *dataglove*, il *Cyberglove 2*. Nel terzo esercizio ai soggetti è stato richiesto di premere diverse combinazioni di dita con forza crescente avvalendosi, per la misurazione, di un dispositivo chiamato *Finger Force Linear Sensor*.

---

<sup>1</sup><http://ninapro.hevs.ch/>

L'attività muscolare viene invece rilevata usando 12 elettrodi wireless, attivi, differenziali doppi, provenienti da un sistema EMG Delsys Trigno Wireless. Gli elettrodi sono disposti in modo da massimizzare i campionamenti mantenendo una distribuzione anatomicamente precisa:

- 8 elettrodi, equidistanti l'uno dall'altro, sono posizionati sull'avambraccio in corrispondenza del giunto radio-omerale;
- 2 elettrodi sono posizionati nei punti di maggior attività dei muscoli flessori ed estensori delle dita (*flexor digitorum* ed *extensor digitorum*);
- 2 elettrodi sono posizionati nei punti di maggior attività dei muscoli bicipite e tricipite.

I segnali sEMG sono stati campionati ad una frequenza di 2 kHz. Durante l'acquisizione, ai soggetti è stato chiesto di ripetere ogni movimento 6 volte, usando la mano destra. Ogni ripetizione è durata 5 secondi, con 3 secondi di riposo tra una ripetizione e l'altra.

Il secondo database conta un totale di 49 movimenti ripetuti 6 volte ciascuno da 40 soggetti sani. Gli esercizi sono stati selezionati sulla base della tassonomia della mano e sulla letteratura riguardante le mani robotiche.

## 2.2 Data Set

Per ogni esercizio e per ogni soggetto, il database fornisce un file .mat che contiene le seguenti variabili, sincronizzate tra loro:

- **subject**: numero del soggetto
- **exercise**: numero dell'esercizio
- **acc**: (36 colonne) rappresenta gli accelerometri a 3 assi dei 12 elettrodi
- **emg**: (12 colonne) segnale sEMG dei 12 elettrodi
- **glove**: (22 colonne) segnale non calibrato dei 22 sensori del cyberglove
- **inclin**: (2 colonne) segnale dell'inclinometro a 2 assi posizionato sul polso
- **stimulus**: (1 colonna) il numero di movimento ripetuto dal soggetto
- **restimulus**: (1 colonna) il numero di movimento ripetuto dal soggetto. In questo caso l'etichetta è ridefinita a posteriori per rispecchiare la corrispondenza con il movimento reale

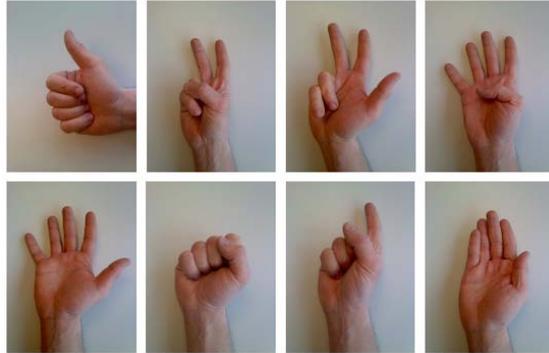


Figura 2.1: Esercizio 1: movimenti isometrici e isotonici della mano

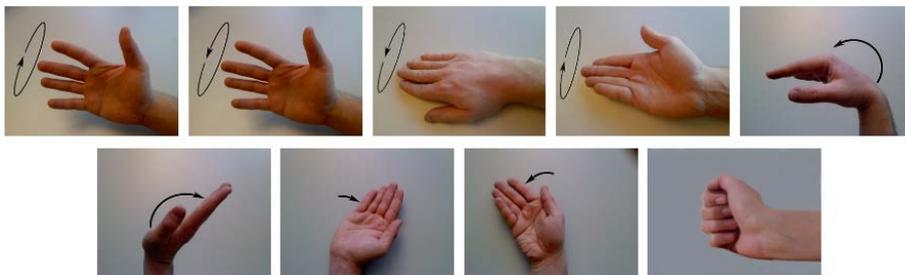


Figura 2.2: Esercizio 1: movimenti basilari del polso



Figura 2.3: Esercizio 2: prese e movimenti funzionali

- **repetition:** (1 colonna) ripetizione di *stimulus*
- **rerepetition:** (1 colonna) ripetizione di *restimulus*
- **force:** (6 colonne) intensità della forza registrata durante il terzo esercizio



Figura 2.4: Esercizio 3: schemi di misurazione della forza per uno o più dita

- **forcecal:** (2 x 6 valori) valori massimo e minimo di calibrazione dei sensori di forza

I segnali *glove* sono quelli provenienti dai sensori del guanto, posizionati come mostrato nella *Figura 2.5*. Nel manuale del CyberGlove è dichiarato che questi dati sono proporzionali agli angoli dei giunti della mano. Inoltre è segnalato che i dati *glove* presenti in Ninapro derivano dall'uso di un guanto non calibrato.

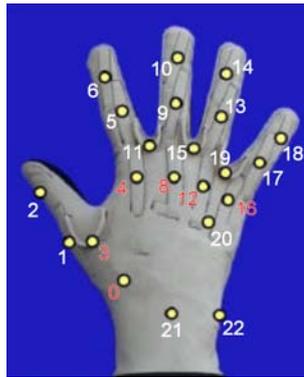
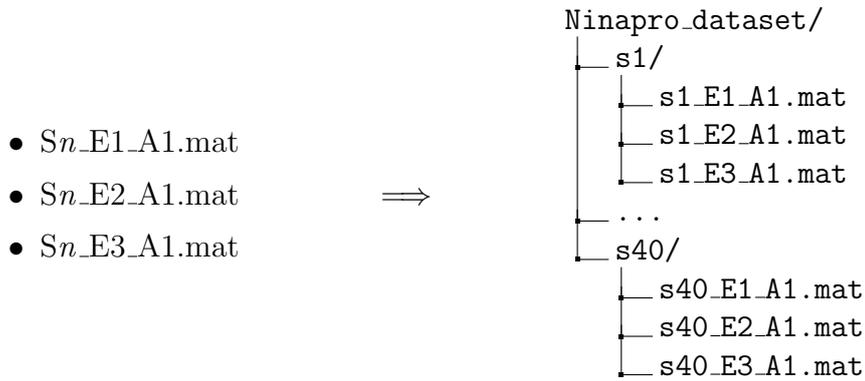


Figura 2.5: Posizionamento dei sensori nel guanto dataglove *Cyberglove 2*.

## 2.3 Estrazione dei dati

Il database si presenta composto da 40 directory, una per soggetto, chiamate  $s1$ , ...,  $s40$ . Ogni directory contiene 3 file con estensione `.mat`, uno per esercizio. Per il soggetto  $n$ -esimo essi sono:



Per ognuno di questi file l'informazione che ci interessa è quella etichettata come *emg* (segnale elettromiografico) e *glove* (cyberglove - movimenti dei giunti della mano). I dati *emg* sono stati usati durante tutto il processo di elaborazione e sono una presenza costante in ogni fase dell'algoritmo. Mentre i segnali *glove* verranno utilizzati solo nella parte finale dell'elaborato, per la costruzione del modello GMM, quindi per il calcolo dell'accuratezza dei canali selezionati.



# Capitolo 3

## Stato dell'arte

### 3.1 Selezione di canali in un sistema sEMG

In letteratura esistono numerosi studi che trattano la classificazione di movimenti e *gesture* tramite l'analisi di segnali biometrici, con una moltitudine di tecniche diverse. Una porzione di questo insieme utilizza dati elettromiografici; ciononostante, solo una parte di questi affronta in maniera attiva il problema della selezione ottimale dei canali in un sistema sEMG. In casi come questi quello che caratterizza maggiormente i risultati è dato da:

- Dati utilizzati
  - Numero di soggetti coinvolti e loro situazione fisico-motoria
  - Arti coinvolti nei movimenti
  - Numero di movimenti considerati
  - Numero di canali sEMG
- Metodi di pre-processing ed elaborazione
- Tecniche di classificazione dei segnali

Per quanto riguarda la tecnica di outlier detection su segnali sEMG della mano, Shibasaki et al. [24] propongono l'utilizzo di una rete neurale probabilistica il cui funzionamento è basato sulla misura di Kullback-Leibler (detta anche divergenza di informazione). Nel loro articolo mettono in pratica questa teoria su un dataset formato da 6 soggetti, uno dei quali ha subito amputazione, lavorando su 16 movimenti totali, misurati da un sistema sEMG a 13 canali. I movimenti trattati nel

loro articolo assomigliano a quelli del dataset Ninapro, come si può notare dalla *Figura 3.1*.

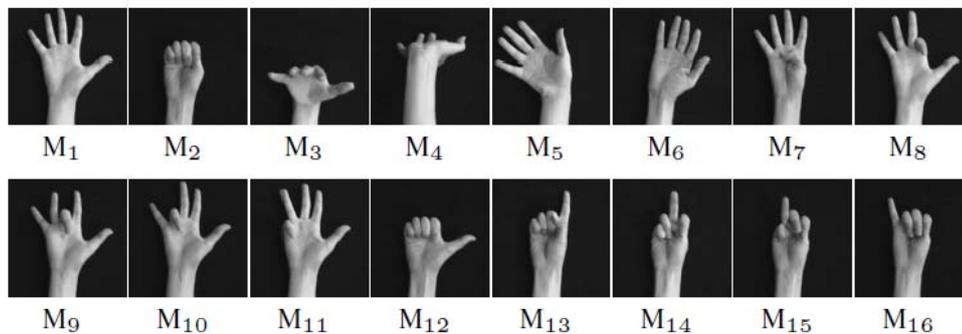


Figura 3.1: Movimenti considerati nell'articolo di Shibasaki et al. [24]

Geng et al. [25] prendono le distanze descrivendo un metodo di selezione dei canali chiamato *multi-class common spatial pattern* (MCCSP) che vanta la peculiarità di essere indipendente dalle tecniche di classificazione usate e dalle feature dei segnali stessi. Questa tecnica è testata su 21 movimenti di 12 soggetti che hanno avuto un trauma cerebrale. La risoluzione è quella di un sistema sEMG a 56 canali. I movimenti considerati sono in parte simili a quelli utilizzati in questo elaborato come si può vedere nella *Figura 3.2*.



Figura 3.2: Movimenti considerati nell'articolo di Geng et al. [25]

Al-Ani et al. [26] sviluppano un algoritmo di selezione dinamico che identifica un sottoinsieme di canali utili per ogni time-segment del segnale, sfruttando un metodo di classificazione dinamica. Lo fanno su 9 soggetti che hanno subito amputazioni, registrando 8 canali sEMG su 6 movimenti: *thumb flexion*, *index flexion*,

*fine pinch, tripod grip, hook grip e spherical grip.*

Infine Liu et al. [27] propongono la riduzione di un insieme di 57 canali sEMG per mezzo di una classificazione basata sui metodi *minimum Redundancy Maximum Relevance* (mRMR) e *Markov random field* (MRF). I segnali in ingresso sono quelli di 9 soggetti colpiti da trauma cervicale spinale incompleto. I movimenti trattati nel loro articolo sono 6 tipologie di prese: *cylindrical grip, tool grip, power grip, lateral pinch, open pinch e fine pinch*. La scelta di è ricaduta su questi movimenti perchè utili nella vita di tutti i giorni.

Si noti quindi che questo elaborato si distacca dagli studi precedenti soprattutto per l'utilizzo di dati provenienti da soggetti sani. Il numero di canali considerati è inferiore, ma è controbilanciato dalla presenza di 6 ripetizioni (*trial*) per ogni movimento e per ogni canale, essenziale per la ricerca di pattern significativi e outlier anche all'interno dell'insieme di segnali dello stesso soggetto. Per quanto riguarda il numero di movimenti, il database Ninapro fornisce un pool di 50 movimenti, nonostante nella fase iniziale di test siano stati considerati solo i primi cinque. Inoltre l'algoritmo che descriveremo in questa tesi espone su una vera e propria classificazione basata sulle feature dei segnali quanto piuttosto un metodo di ranking dei canali che misura la loro somiglianza utilizzando una metrica tipica dell'analisi dei segnali che è il concetto di correlazione/covarianza incrociata, affinata in alcuni casi con la differenza tra i segnali.

## **3.2 Modelli probabilistici per il movimento di protesi robotiche a partire da segnali biometrici**

In letteratura sono numerosi gli studi che trattano come classificare il movimento di un arto, altrettanto numerosi sono gli articoli che mirano a risalire all'esatta posizione dell'arto partendo dai segnali EMG associati a quel movimento. Le tecniche utilizzate per risolvere questi task sono spesso basate su modelli probabilistici i quali, una volta allenati con un training set, sono in grado di decifrare i movimenti associati ad un test set di segnali biometrici. I modelli utilizzati e proposti sono di vario tipo: Neural Networks (NN) [28], Gaussian Mixture Model (GMM) [41], Hidden Markov Model (HMM) [42]. Nel loro articolo Scheme et al.

[43] descrivono un gran numero di tecniche probabilistiche confrontandole sotto diversi punti di vista. Huang et al. [41] sottolineano che nell'affrontare problemi di questo tipo vanno presi in considerazione numerosi parametri per ottenere una valutazione il più completa ed oggettiva possibile: accuratezza, efficienza, capacità di adattamento a nuovi dati, ottimizzazione rispetto a pattern unici, quantità di dati di training utilizzati per allenare il modello. Concludono asserendo che il modello GMM le soddisfa tutte se utilizzato per fini di classificazione. Gli articoli più recenti che approfondiscono questo argomento invece mirano a determinare il movimento completo dell'un arto piuttosto che a classificarlo. Han et al. [44] hanno sviluppato un modello *emg* basato sul modello Hill Muscle per la stima continua del giunto del gomito. Tuttavia il loro modello richiede una serie di parametri fisiologici che ne inficiano la generalità. Inoltre le sue prestazioni sono limitate dalla complessità computazionale che ne preclude l'utilizzo in applicazioni real time. Al contrario Manal et al. [45] hanno descritto un braccio virtuale che riproduce movimenti in tempo reale basandosi sul modello Hill Muscle e su segnali EMG. In maniera simile Chen et al. [46] nel loro articolo propongono la stima in tempo reale dei movimenti del gomito sfruttando un algoritmo di Hierarchical Projected Regression (HPR) che costruisce in maniera incrementale una libreria ad albero le cui varie componenti sono modelli di regressione locali.

Michieletto et al. in [47] trattano l'argomento in maniera più aderente a quanto fatto in questo elaborato, costruendo un framework probabilistico basato su GMM per la stima di un giunto basandosi su più canali di un sistema sEMG. Il segnale EMG viene analizzato nella sua interezza ottenendo risultati molto buoni per ognuno dei soggetti coinvolti nello studio, limitandosi all'applicazione offline. In un lavoro successivo Valentini et al. [30] stimano in tempo reale la posizione di un giunto grazie ad un modello GMM allenato con feature EMG estratte utilizzando la trasformata wavelet (vedi *sezione 4.1*). Questa trasformata permette di considerare solo una finestra limitata dell'intero segnale a partire dalla quale, per mezzo di un algoritmo GMR, viene stimato l'angolo del giunto del ginocchio. Vengono mantenute specificità rispetto al soggetto ed efficienza della computazione, grazie alla velocità del modello GMM/GMR. Mentre in [31] viene utilizzato un approccio indipendente dal soggetto. Inoltre il numero di parametri richiesti sono limitati, il che si traduce in un framework leggero.

# Capitolo 4

## Pre-processing

### 4.1 Trasformata Wavelet

Le Wavelet sono un tipo di trasformata che consente un'analisi nei tempi e nelle frequenze. Applicando al nostro segnale una trasformata Wavelet potremo ottenere un'elaborazione online in quanto la trasformata wavelet continua o Continuous Wavelet Transform (CWT) di un segnale  $x(t)$  è definita come:

$$W(s, \tau) = \int x(t)\psi_{s,\tau}^*(t)dt \quad (4.1)$$

La wavelet madre (o *mother wavelet*)  $\psi$  è una funzione tale che:

$$\int_{-\infty}^{+\infty} \psi(t)dt = 0 \quad (4.2)$$

$\psi$  è scalata secondo il parametro  $s$ , che misura il grado di compressione, e traslata secondo il parametro  $\tau$ , che ne determina il posizionamento sull'asse temporale. Agendo su  $s$  e  $\tau$  quindi, vengono fornite informazioni del segnale a tempi e frequenze differenti.

La *mother wavelet* scalata e traslata è definita come:

$$\psi_{s,\tau}(t) = \frac{1}{\sqrt{s}}\psi\left(\frac{t-\tau}{s}\right) \quad (4.3)$$

Combinando *Equazione 4.1* e *Equazione 4.3*, la wavelet transform per il segnale  $x(t)$  può essere scritta come:

$$W(s, \tau) = \frac{1}{\sqrt{s}} \int x(t) \psi\left(\frac{t - \tau}{s}\right) dt \quad (4.4)$$

La trasformata wavelet discreta (*Discrete Wavelet Transform* o *DWT*)  $f[n]$  di un segnale discreto appartenente all'insieme  $l^2(Z) = \{f[n] \mid \sum_{n=-\infty}^{\infty} |f[n]|^2 < \infty\}$  è definita come:

$$f[n] = \frac{1}{\sqrt{M}} \sum_k W_\phi[j_0, k] \phi_{j_0, k}[n] + \frac{1}{\sqrt{M}} \sum_{j=j_0}^{\infty} \sum_k W_\psi[j, k] \psi_{j, k}[n] \quad (4.5)$$

dove  $\phi(t)$  è la *scaling function*,  $\psi(t)$  è la *wavelet function*,  $j$  rappresenta il livello di decomposizione e  $M$  il numero di punti. Dal momento che gli insiemi  $\{\phi_{j_0, k}[n]\}_{k \in Z}$  e  $\{\psi_{j, k}[n]\}_{(j, k) \in Z^2, j \geq j_0}$  sono ortogonali fra loro [32], è possibile usare il prodotto scalare per ottenere i *wavelet coefficients*:

$$W_\phi[j_0, k] = \frac{1}{\sqrt{M}} \sum_n f[n] \phi_{j_0, k}[n] \quad (4.6)$$

$$W_\psi[j, k] = \frac{1}{\sqrt{M}} \sum_n f[n] \psi_{j, k}[n] \quad j \geq j_0 \quad (4.7)$$

che rappresentano rispettivamente i coefficienti approssimati (*Equazione 4.6*) e i coefficienti dettagliati (*Equazione 4.7*) della trasformata wavelet.

L'efficacia della wavelet transform nell'estrarre delle feature appropriate dipende fortemente dalla scelta della *mother wavelet* usata per i calcoli. Anche se applicate allo stesso segnale, diverse mother wavelet possono estrarre feature molto differenti. Alcune delle famiglie principali di mother wavelet sono Haar, Daubechies, Coiflet, Symmlet, Morlet e Mexican Hat. Anche se non esiste una regola specifica per la selezione della funzione migliore, alcune mother wavelet sono più

adatte di altre per determinati tipi di segnali.

I segnali sEMG registrati sono estremamente variabili e poco caratterizzanti. Il loro utilizzo senza una fase di preprocessing li rende pressochè inutilizzabili, si rende quindi necessario applicare un preprocessing per evidenziare le caratteristiche peculiari del segnale. Lo strumento utilizzato per l'estrazione di *feature*, come suggeriscono alcuni studi a riguardo [29], è la trasformata Wavelet o *Wavelet Transform*. Il suo utilizzo porta con se dei vantaggi tra cui l'esclusione del tempo dall'analisi dei segnali. Ciò consente di ridurre i tempi di calcolo fino al punto di poter eseguire un'elaborazione online, come viene riportato da Valentini et al. in [30].

Il preprocessing è stato eseguito grazie ad una libreria C++, *wavelib*, che calcola la trasformata wavelet discreta o Discrete Wavelet Transform (DWT) del segnale sEMG grezzo, generando i file con suffisso *\_wlt*. I parametri utilizzati, suggeriti da studi precedenti [30][31], sono i seguenti:

- *tipo trasformata*: Discrete Wavelet Transform
- *mother wavelet*: Daubechies 2 (db2)
- *decomposition level*: 1
- *wlt window length*: 200 msec

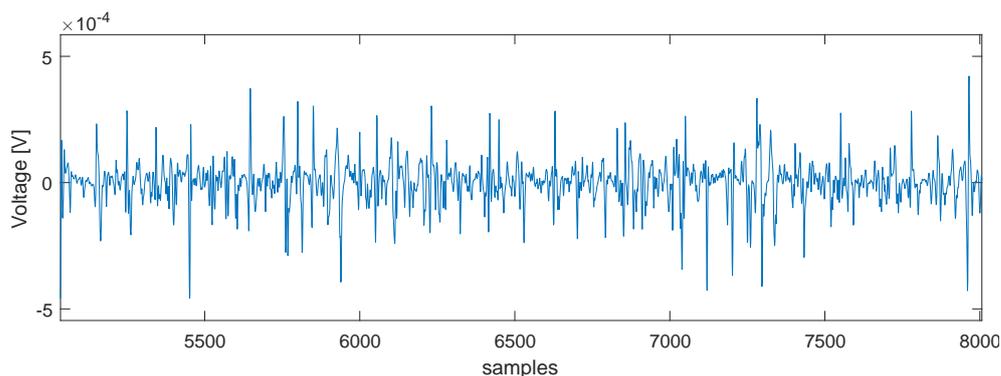


Figura 4.1: Segnale sEMG grezzo, relativo ad un movimento, nella sua interezza.

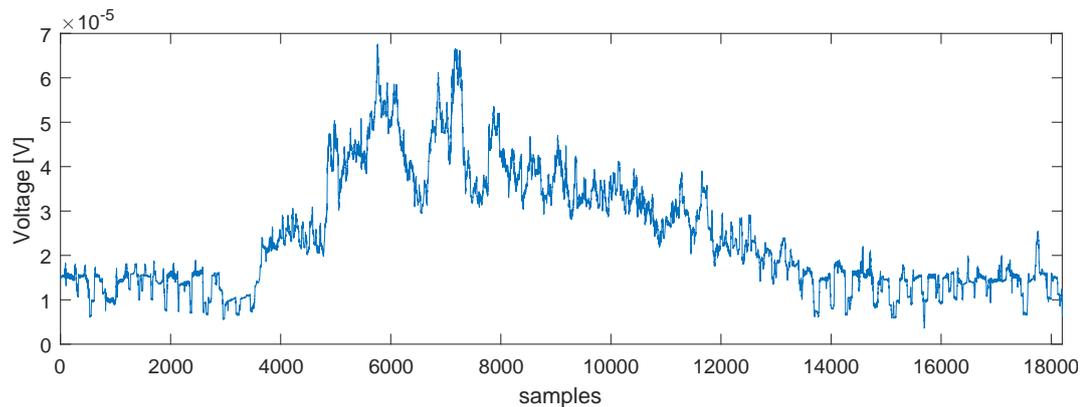


Figura 4.2: Segnale sEMG dopo essere stato elaborato con Discrete Wavelet Transform

## 4.2 Elaborazione dei segnali

Una volta estratti i dati d'interesse dal database e applicata la trasformata wavelet, i segnali EMG sono stati elaborati usando Matlab al fine di metterne in evidenza i tratti distintivi, aumentandone così la riconoscibilità dei pattern. In particolare, al segnale grezzo, sono state applicate diverse combinazioni delle seguenti elaborazioni:

- **Smoothing:** funzione Matlab che filtra il segnale con lo scopo di evidenziare i pattern significativi attenuando il rumore. Viene eseguito con metodo *loess*, ovvero con una regressione locale usando il metodo dei minimi quadrati lineare e pesato, assieme ad un modello polinomiale di secondo grado.
- **Normalizzazione:** rimappa il supporto dell'intero segnale tra zero e uno. Per fare ciò viene usata la formula:

$$signal_{normalized} = \frac{signal - min(signal)}{max(signal) - min(signal)} \quad (4.8)$$

- **Traslazione verticale:** fa sì che tutti i segnali partano dallo stesso punto, diminuendo, di conseguenza, la differenza assoluta tra i segnali relativi a *trial* differenti dello stesso movimento.

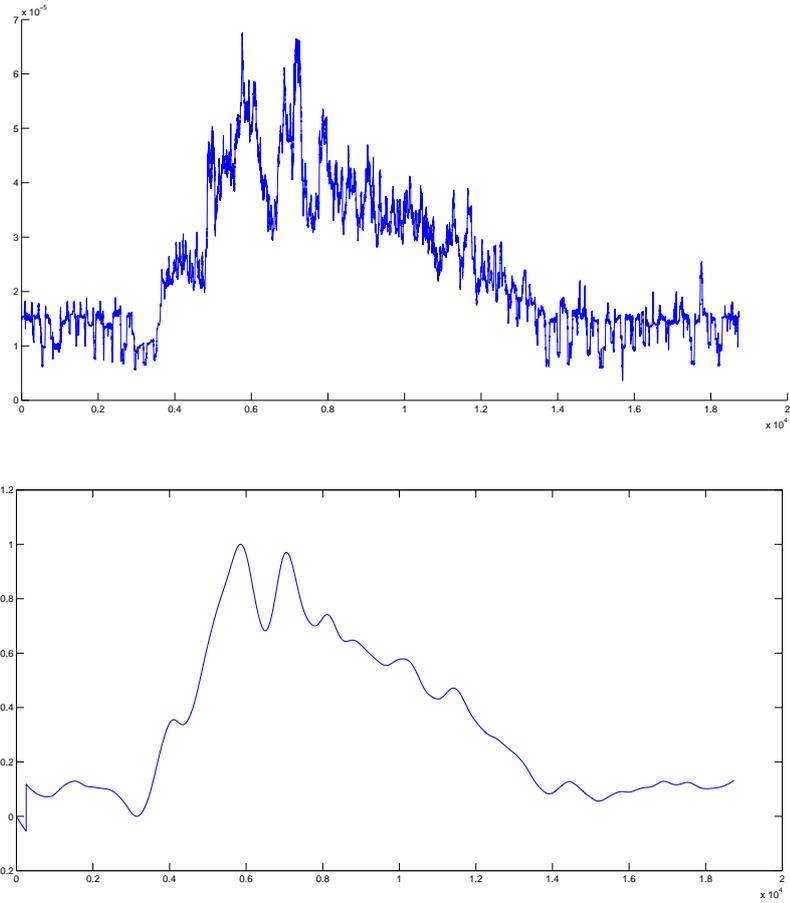


Figura 4.3: Stesso segnale DWT prima e dopo smoothing, normalizzazione e traslazione



# Capitolo 5

## Outliers detection

Rappresentando graficamente alcuni trial (vedi *Figura 5.1*) è facile accorgersi come i segnali EMG per uno stesso movimento possano essere molto diversi tra loro, anche considerando un solo soggetto. Questo solleva la necessità di predisporre una fase di rimozione dei trial che si discostano da quello che è l'andamento generale della maggior parte dei segnali, considerando un canale per volta.

È doveroso notare che il riconoscimento di outlier all'interno di un pool di segnali dei quali non si conosce a priori la classe (segnale normale o segnale outlier) è da considerarsi un task di *unsupervised learning* o nella fattispecie di *unsupervised anomaly detection* [19]. Questo ci costringe ad adottare una tecnica *ensemble* di selezione a maggioranza: per ogni movimento e per ogni canale vengono ricercati gli outlier tra i 6 trial di tutti e 40 i soggetti con  $n$  metodi diversi. Se un particolare trial è segnalato dalla maggioranza dei metodi utilizzati esso viene etichettato come outlier ed escluso dall'algoritmo nella fase successiva. I metodi di anomaly detection sono stati selezionati sulla base delle loro complessità computazionale per mantenere delle prestazioni efficienti, e sulla loro adattabilità al contesto dell'analisi di segnali [20][21][22][23].

Di seguito i metodi di outlier detection utilizzati.

### 5.1 Hampel filter

Il filtro mediano è basato su una finestra mobile (*sliding window*) simmetrica il cui unico parametro è dato dalla dimensione della finestra stessa. Il filtro di Hampel

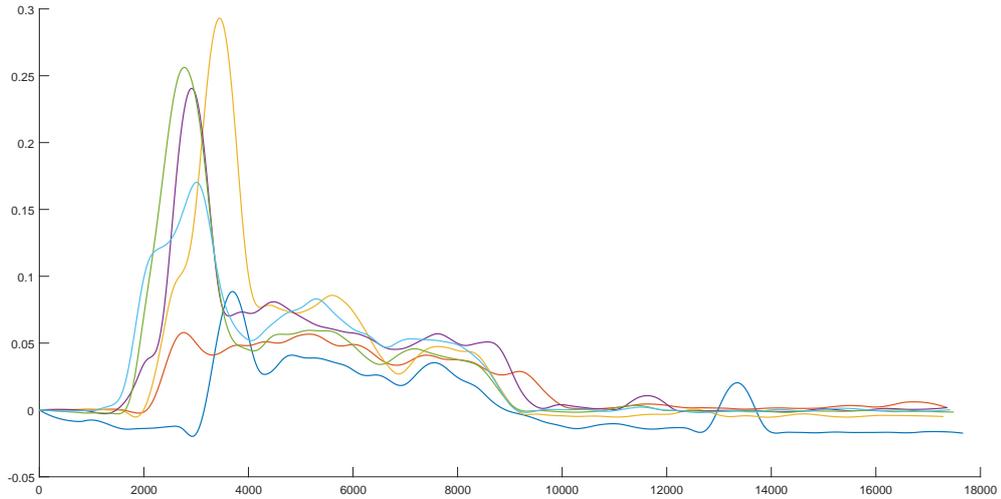
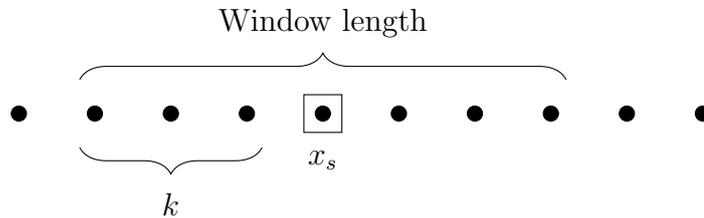


Figura 5.1: I 6 trial per Movimento M2, Soggetto S1, canale 4, normalizzati per canale e per movimento.

basa il suo funzionamento su un concetto simile. Esso sostituisce il valore centrale della finestra mobile con la mediana dei valori sui quali essa spazia, nel caso in cui questo sia sufficientemente lontano dalla mediana da essere etichettato come outlier [9].



In maniera più formale, data una sequenza  $x_1, x_2, x_3, \dots, x_n$ , la finestra mobile che scorre sul segnale è indicata come:

$$W_k^K = \{x_{k-K}, \dots, x_k, \dots, x_{k+K}\} \quad (5.1)$$

dove  $K$  è un intero positivo che rappresenta la lunghezza della finestra mobile. Il filtro mediano  $M_K$  definito sulla finestra  $W_k^K$  sostituisce ad ogni valore del segnale  $x_k$  la mediana dei campioni contenuti nella finestra  $m_k$ :

$$x_k \leftarrow m_k = \text{median}(x_{k-K}, \dots, x_k, \dots, x_{k+K}) \quad (5.2)$$

rendendo questo tipo di filtro dipendente dalla sola lunghezza della finestra  $K$ .

Il filtro di Hampel  $H_K$  elabora questo concetto aggiungendo un intervallo entro il quale ogni campione viene considerato outlier e quindi sostituito dalla mediana della sliding window centrata su di esso. Si tratta, in effetti, di una implementazione dell'*Hampel identifier* descritto da Davies et al. [10], basata sul valore di Mean Absolute Deviation (MAD) come stima della deviazione standard per dati con distribuzione normale. Nello specifico la risposta del filtro rispetta la funzione [3]:

$$y_k = \begin{cases} x_k, & \text{if } |x_k - m_k| \leq tS_k \\ m_k, & \text{if } |x_k - m_k| > tS_k \end{cases} \quad (5.3)$$

dove  $S_k$  è la stima della deviazione standard e  $t$  è l'altro parametro, oltre alla lunghezza della sliding window  $K$ , da cui dipende  $H_K$ . Inoltre vale che:

$$S_k = \xi \times m_k \quad (5.4)$$

$$\xi = \frac{1}{\sqrt{2} \operatorname{erfc}^{-1}(1/2)} \approx 1.253 \quad (5.5)$$

Moltiplicare per il fattore 1.253 permette di ridurre il *bias* tra la stima fatta dalla MAD e il reale valore assunto dalla deviazione standard, sotto l'ipotesi che i dati presi in considerazione abbiano una distribuzione gaussiana.

Il concetto, definito per il singolo campione, è stato esteso agli interi segnali considerando outlier un qualsiasi trial il quale conti più di una certa percentuale  $p\%$  di campioni outlier.

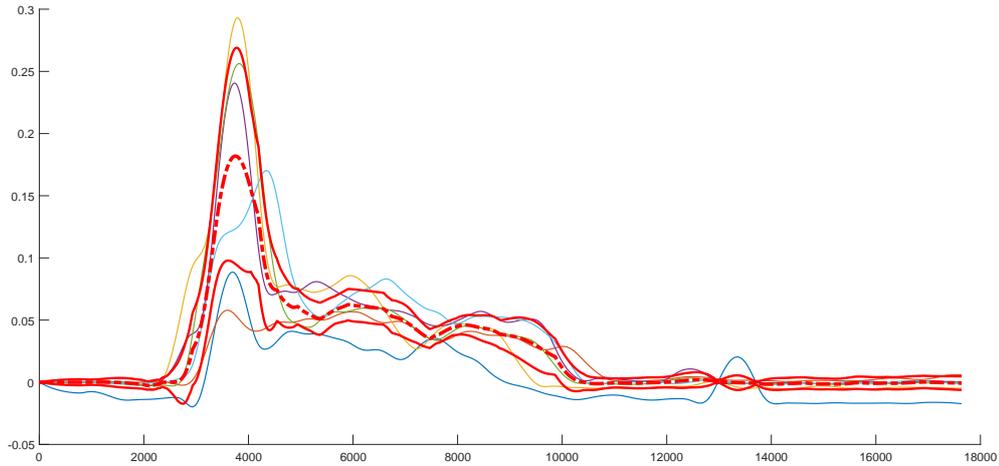


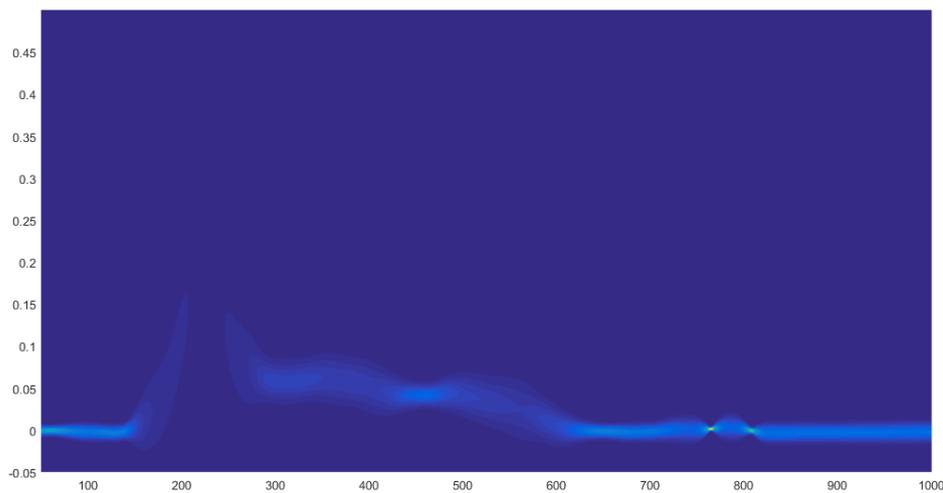
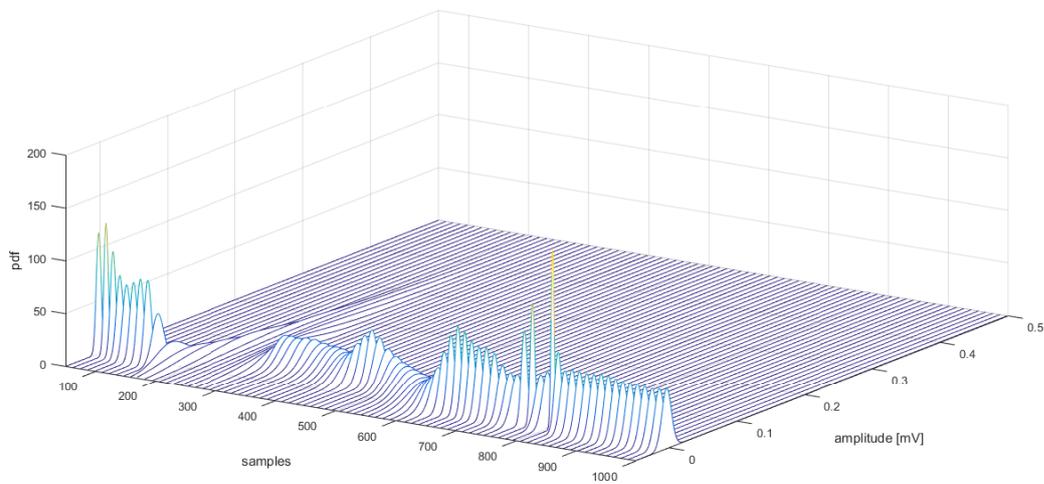
Figura 5.2: Rispetto alla *Figura 5.1*, in rosso tratteggiato è evidenziata la mediana dei segnali per ogni campione. Il tratto rosso continuo indica invece la *mean absolute deviation*.

## 5.2 Gaussian distribution fitting

Ogni segnale del dataset è assimilabile ad una *time series* in quanto esso rappresenta la ripetizione di un movimento da parte di un soggetto per un determinato canale. Consideriamo un dato istante di campionamento come un evento del quale sono state osservate un numero di realizzazioni pari a *numero soggetti*  $\times$  *numero trial* =  $40 \times 6 = 240$ . Con un tale numero di osservazioni e sotto l'ipotesi di normalità della distribuzione che modella le osservazioni di questo esperimento, è lecito pensare di costruire una variabile aleatoria gaussiana che descriva nel miglior modo quei dati. L'idea di base rappresenta una variazione del più classico utilizzo di diagrammi a scatola (boxplot) [16] per lo studio della distribuzione di alcuni campioni tramite semplici indici di dispersione e di posizione [11].

Più formalmente, il segnale sEMG è visualizzato come un insieme di variabili aleatorie normali  $\{X_k(t)\}_{k=1,\dots,n}$ , una per ogni istante di campionamento, dove  $n$  è il numero di campioni del segnale. Per ogni istante di campionamento considero  $m$  segnali differenti, quindi  $m$  diversi campioni provenienti da altrettanti trial, sulla base dei quali generare una distribuzione gaussiana. Ciò che si ottiene sono  $n$  variabili normali per le quali calcolare un intervallo di confidenza che discriminerà se un certo campione è outlier o meno. In maniera analoga a quanto descritto nella *sezione 5.1*, il concetto è stato esteso all'intero segnale considerando outlier

un qualsiasi trial il quale conti più di una certa percentuale  $p\%$  di campioni outlier.



## 5.3 Outliers detection basata su PCA

### 5.3.1 Principal Component Analysis (PCA)

La Principal Component Analysis (PCA), o analisi in componenti principali, è una tecnica per la semplificazione dei dati utilizzata nell'ambito della statistica multivariata per estrarre da un set di dati nuovi attributi che meglio rappresentino la

variabilità dei dati. Fu proposta nel 1901 da Karl Pearson [48] e sviluppata da Harold Hotelling nel 1933 [49][50]. È nota anche come trasformata di Karhunen-Love (KLT), trasformata di Hotelling o decomposizione ortogonale propria (POD, dall'inglese *proper orthogonal decomposition*).

Questa tecnica ha come scopo primario quello di ridurre un certo numero di variabili, le quali rappresentano le caratteristiche del fenomeno analizzato, in alcune variabili latenti eseguendo così una *feature reduction*. Nello specifico la prima dimensione è scelta per catturare più variabilità possibile. La seconda dimensione assomiglia a sua volta più variabilità possibile, sotto l'ipotesi di ortogonalità rispetto alla prima dimensione. Il ragionamento prosegue invariato per le dimensioni successive.

Le caratteristiche più promettenti della PCA per le applicazioni di *dimensionality reduction* sono rappresentate in primo luogo dalla sua capacità di identificare i pattern più robusti e significativi contenuti all'interno dei dati. In secondo luogo la maggior parte di variabilità viene solitamente catturata all'interno delle prime dimensioni, consentendo una riduzione notevole delle dimensioni del problema considerato e rendendo possibile l'applicazione di metodi che funzionano meglio con dati a poche dimensioni [19].

Ciò avviene tramite una trasformazione lineare delle variabili che proietta quelle originarie in un nuovo sistema cartesiano nel quale la nuova variabile con la maggiore varianza viene proiettata sul primo asse, la seconda per dimensione della varianza sul secondo asse e così via. La riduzione della complessità avviene limitandosi ad analizzare le principali (per varianza) tra le nuove variabili.

A differenza di altre trasformazioni lineari adoperate in statistica, nella PCA i vettori di trasformazione sono generati sulla base dei dati di partenza. Sotto l'ipotesi che a ciascuna delle variabili originarie venga sottratta la loro media, ottenendo così una nuova variabile  $x_i$  con media nulla, possiamo calcolare la prima componente principale come:

$$\mathbf{w}_1 = \arg \max_{\|\mathbf{w}\|=1} \left\{ (\mathbf{w}^T \mathbf{x})^2 \right\} \quad (5.6)$$

(Dove *argmax* indica l'insieme degli argomenti  $w$  in cui è raggiunto il massimo.)  
 Conoscendo i primi  $k - 1$  componenti, il  $k - \text{esimo}$  componente può essere trovato sottraendo i primi  $k - 1$  componenti principali a  $X$ :

$$\hat{\mathbf{x}}_{k-1} = \mathbf{x} - \sum_{i=1}^{k-1} \mathbf{w}_i \mathbf{w}_i^T \mathbf{x} \quad (5.7)$$

e sostituendo si ottiene:

$$\mathbf{w}_k = \arg \max_{\|\mathbf{w}\|=1} E \left\{ (\mathbf{w}^T \hat{\mathbf{x}}_{k-1})^2 \right\} \quad (5.8)$$

Nel corso di questo elaborato verranno utilizzate le prime due componenti del punteggio della PCA per poi essere sottoposte ad un'analisi ulteriore per la ricerca di outlier (vedi *Capitolo 5*).

### 5.3.2 Robust multivariate covariance estimate

Quando si ha a che fare con la ricerca di outlier [18] all'interno di un database o più in generale con l'analisi esplorativa di alcuni dati (*exploratory data analysis*), è pratica comune quella di ricorrere all'utilizzo di misure statistiche cosiddette robuste (*robust statistic*). Una procedura statistica, o più semplicemente una statistica, viene definita robusta (o resistente) se il valore della stima utilizzata non è sensibile a piccole variazioni dei dati presi in considerazione [12], dove con piccole variazioni dei dati si intendono piccole variazioni in tutti i dati oppure grandi variazioni solo per alcuni dati.

Nell'ambito dell'analisi di dati multivariati (i.e. descritti e/o definiti su un numero di dimensioni  $n > 2$ ) le statistiche robuste ricoprono un ruolo significativo. Vengono spesso utilizzate stime robuste della (matrice di) covarianza per ottenere una semplice descrizione dell'ellissoide che racchiude la nuvola di punti formata da dati  $n$ -dimensionali [12].

Inoltre l'applicazione di stime robuste della covarianza, congiuntamente a tecniche di analisi dei dati come la PCA, viene spesso sfruttata per ottenere una riduzione delle dimensioni (*dimensionality reduction*) sui dati da studiare [12][13],

come già descritto nella *sottosezione 5.3.1*.

Nel contesto di questa tesi viene applicata l'analisi PCA ai segnali sEMG, che in questo frangente sono visti come dei punti con tante dimensioni quanti sono il numero di campioni di ogni segnale. Lo scopo, come illustrato poc'anzi, è quello di eseguire una dimensionality reduction: si passa dalle  $n$  dimensioni del trial, date dal numero di campioni, alle 2 dimensioni descritte dalle prime 2 componenti del punteggio di un'analisi PCA.

A questo punto ogni trial viene "proiettato" in uno spazio bidimensionale, all'interno del quale eseguire una stima robusta della covarianza per cercare eventuali outlier. Si noti tuttavia che questo tipo di stima può essere calcolata in diversi modi. Ne sono stati considerati 3, che figureranno come 3 diversi outlier detector nel metodo *ensemble* descritto a inizio del *Capitolo 5*. Essi sono:

- *Minimum Covariance Determinant (MCD) Estimate*

È un tipo di stima applicabile ai casi multivariati [18], con le proprietà di essere consistente e robusto. Più precisamente esso è definito nel modo seguente. La prima fase per il calcolo della stima MCD per un insieme di  $n$  punti consiste nel trovare un sottoinsieme di  $h$  punti, con  $\frac{n}{2} < h < n$ , che minimizzi la varianza calcolata su quel sottoinsieme (cioè il determinante della matrice di covarianza calcolata sugli  $h$  punti). Quindi le stime di media e varianza sono date dalla media e dalla varianza del sottoinsieme ottimale [14].

Un'esatta valutazione di MCD è computazionalmente impraticabile, infatti sarebbe troppo dispendioso in termini di tempo valutare tutti i possibili sottoinsiemi dei dati in ingresso. Per questo motivo viene usato l'algoritmo Fast Minimum Covariance Determinant (FMCD) descritto da Rousseuw et al. [6], al quale sono stati apportati opportuni fattori correttivi con lo scopo di ridurre il bias introdotto da insiemi di campionamento molto piccoli [7].

- *Orthogonalized Gnanadesikan-Kettenring (OGK) Estimate*

L'algoritmo OGK implementa una stima per dati multivariati partendo dalla stima di Gnanadesikan and Kettenring (GK) che genera una matrice robusta che potrebbe però non essere definita positiva [8]. Questa stima usa una forma di componente principale eseguendo un'ortogonalizzazione su tale matrice, sostituendo i suoi eigenvalues, che potrebbero essere negativi, con

varianze robuste. Questa procedura viene reiterata per migliorare il risultato e converge solitamente nel giro di 2 o 3 iterazioni.

- *Olive Hawkins Estimate*

La di stima di Olive-Hawkins usa l’algoritmo di concentrazione proposto da Olive-Hawkins [4]. Si tratta di una classe di algoritmi veloci, consistenti e con la proprietà di essere fortemente resistente e contro gli outlier [5]. La covarianza viene stimata in maniera robusta secondo un metodo quadratico che sfrutta 4 momenti.

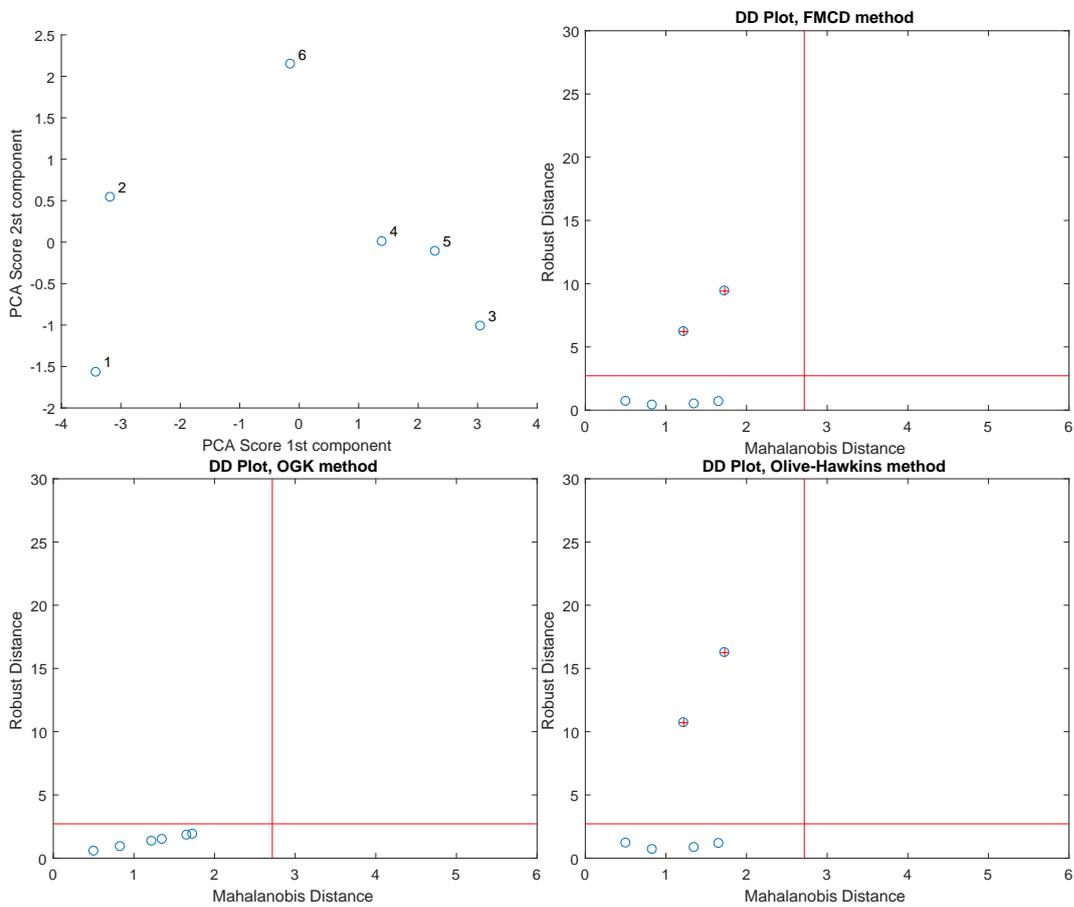


Figura 5.3: caption

Un metodo conveniente, spesso utilizzato per la visualizzazione dei risultati della stima della covarianza è il grafico DD-plot (Distance-Distance o Depth-Depth

plot). Questo tipo di grafico viene spesso usato per l'analisi preventiva di dati non conosciuti per confrontare distribuzioni bivariate basandosi sulla distanza tra dati [51]. In *Figura 5.3* vengono confrontate di Mahalanobis con quella robusta, generata nel calcolo della stima di covarianza.

### 5.3.3 Bivariate gaussian distribution fitting

In questo metodo si punta a costruire una variabile aleatoria normale bivariata che approssimi al meglio i dati composti dalle prime due componenti del risultato di un'analisi PCA [17]. Una volta trovata la funzione densità di probabilità si fissa un intervallo di confidenza fuori dal quale un campione è considerato outlier [19].

Siano  $x$  e  $y$  la prima e la seconda componente del punteggio dell'analisi PCA, rispettivamente. Assumiamo che esse siano assimilabili a variabili normali ovvero che  $x \sim N(\mu_x, \sigma_x)$  e  $y \sim N(\mu_y, \sigma_y)$ . Allora vale:

$$p(x, y) = \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}} e^{-\frac{1}{2(1-\rho^2)}Q(x,y)} \quad (5.9)$$

dove la forma quadratica  $Q(x, y)$  è data da:

$$Q(x, y) = \left(\frac{x - \mu_x}{\sigma_x}\right)^2 - 2\rho\left(\frac{x - \mu_x}{\sigma_x}\right)\left(\frac{y - \mu_y}{\sigma_y}\right) + \left(\frac{y - \mu_y}{\sigma_y}\right)^2 \quad (5.10)$$

$p(x, y)$  è la funzione densità di probabilità della variabile normale bivariata e dipende da 5 parametri:  $\mu_x, \mu_y, \sigma_x, \sigma_y, \rho$ , per i quali valgono le seguenti relazioni:

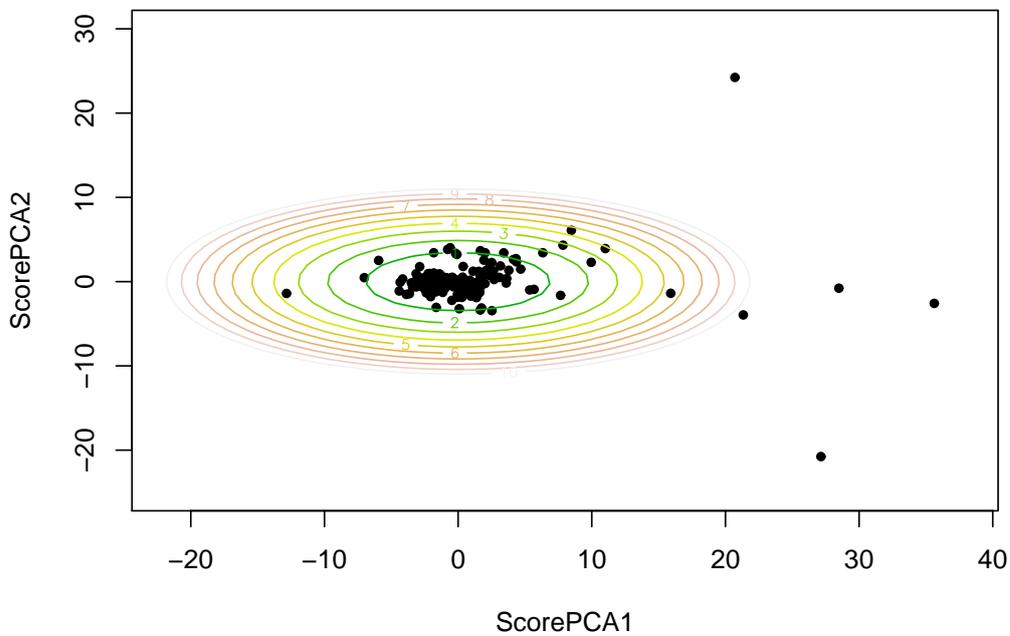
$$E(x) = \mu_x \quad E(y) = \mu_y \quad var(x) = \sigma_x^2 \quad var(y) = \sigma_y^2 \quad (5.11)$$

mentre il parametro  $\rho$  è il coefficiente di correlazione e soddisfa:

$$E[(x - \mu_x)(y - \mu_y)] = \rho\sigma_x\sigma_y \quad (5.12)$$

Visivamente  $\rho$  indica quanto la curva normale bivariata è "schiacciata" (per valori di  $|\rho| \rightarrow 1$  si otterrà un grafico molto "stretto") e in quale direzione è orientata. Per valori di  $\rho > 0$  la variabile  $y$  tende ad aumentare all'aumentare di  $x$ . Viceversa per valori di  $\rho < 0$  ad un incremento di  $x$  corrisponderà una diminuzione di  $y$ . Si noti che quando  $\rho = 0$  allora  $x$  e  $y$  sono indipendenti. La quantità  $E[(x - \mu_x)(y - \mu_y)]$  rappresenta la covarianza delle variabili  $x$  e  $y$  e viene spesso indicata con  $cov(x, y)$ . Infatti, date due variabili qualsiasi si definisce:

$$cov(x, y) = E[(x - E(x))(y - E(y))] \quad (5.13)$$





# Capitolo 6

## Processo di selezione

### 6.1 Scelta preliminare dei canali

Fissati un soggetto, un movimento e un canale, viene analizzata graficamente l'evoluzione nel tempo dei 6 *trial*, elaborati secondo quanto scritto nella *sezione 4.2* (e.g. *Figura 6.1* e *Figura 6.2*). Da qui si procede con la scelta manuale dei canali più promettenti basandosi sull'osservazione empirica. Consideriamo degno di nota un canale che soddisfi le seguenti caratteristiche:

- Presenti grosso modo lo stesso andamento per ognuno dei 6 *trial*: questo vuol dire che in una successiva ripetizione di quel movimento ci si aspetta di ottenere un *pattern* simile cioè un movimento riconoscibile se ripetuto in futuro. Questo si traduce in riconoscibilità e ripetibilità del segnale.
- Sia caratterizzato da un andamento poco altalenante.
- Non presenti eccessiva differenza tra i vari *trial* in termini di valori assunti dal segnale.

In questo frangente la scelta manuale serve per avere un risultato iniziale dal quale partire per discriminare o confermare i futuri risultati che saranno il frutto di un algoritmo preciso.

All'atto pratico, come prima prova, è stato scelto un sottoinsieme composto da 20 soggetti per i quali testare empiricamente la strategia appena illustrata. Considerando a titolo di esempio il movimento M1, sono stati osservati uno alla volta i grafici dei 12 canali, soggetto per soggetto, e tra questi sono stati selezionati,

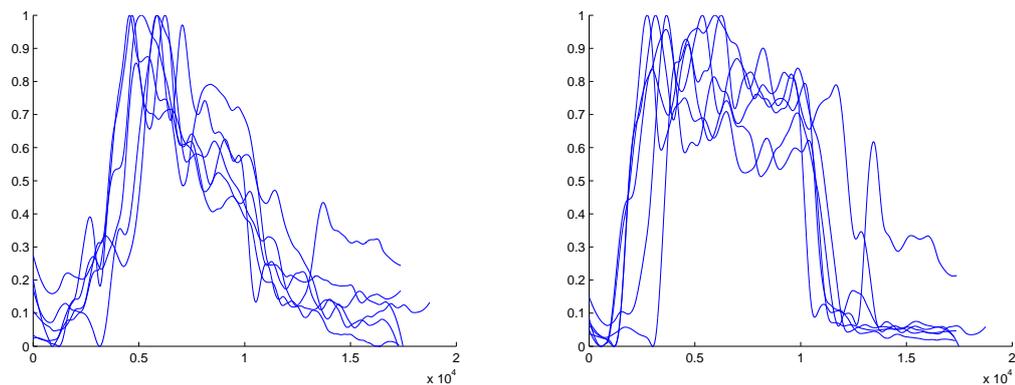


Figura 6.1: Esempio di canali potenzialmente utili: i 6 trial mostrano un'andamento simile. (Esercizio 1, Movimento 1, Soggetto 1, Canali 1 e 2)

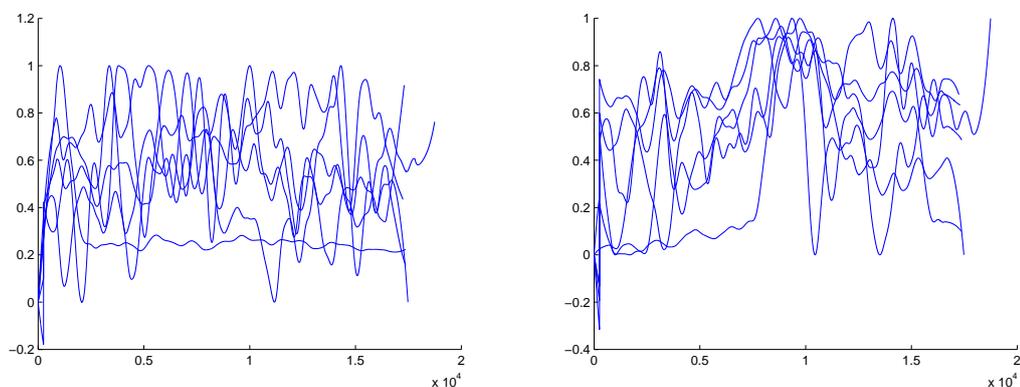


Figura 6.2: Esempio di canali scartabili: i 6 trial mostrano andamenti altalenanti e diversi tra loro. (Esercizio 1, Movimento 1, Soggetto 1, Canali 11 e 12)

osservandoli, i 3 migliori canali che rispettassero i criteri di bontà descritti poc'anzi. Il risultato ottenuto alla fine di questo processo è rappresentato in *Tabella 6.1* e in *Figura 6.3*.

S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
1	8	3	2	1	2	1	2	3	1
2	9	7	4	2	3	2	5	8	2
5	10	10	10	8	10	8	8	11	8
S11	S12	S13	S14	S15	S16	S17	S18	S19	S20
1	2	1	1	2	1	1	1	5	3
2	4	2	2	3	2	2	2	9	8
10	9	8	3	8	8	10	8	10	9

Tabella 6.1: Risultato della selezione dei canali significativi con metodo dell'osservazione empirica a votazione. La tabella racchiude i risultati per il movimento M1 con test su un campione di 20 soggetti da S1 a S20. I canali più promettenti sono risultati il canale 1, 2 e 8 perchè sono quelli comparsi più frequentemente nei top-3 analizzando soggetto per soggetto

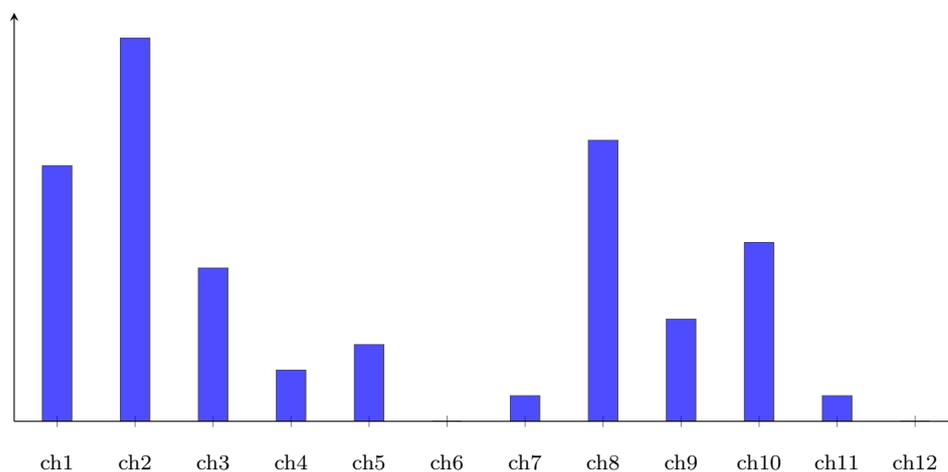


Figura 6.3: Istogramma rappresentativo della *Tabella 6.1*. Si noti come spiccano fra gli altri i canali 1, 2 e 8 nonostante la popolazione sia di soli 20 soggetti sui 40 totali

Ricapitolando: sono stati scelti 3 canali per ogni movimento. Il metodo usato per la scrematura è stato quello dell'osservazione qualitativa: sono stati confrontati i grafici dei canali uno alla volta e selezionati quelli che avevano andamenti simili per i vari trial. Poi conteggiati quelli che, soggetto per soggetto, comparivano con maggior frequenza.

Ribadiamo tuttavia che l'osservazione empirica della "somiglianza" dei trial non può essere considerato un metodo preciso e obiettivo, in quanto varia da osservatore a osservatore. Si è cercato quindi di adottare un metodo sistematico per la selezione dei canali promettenti che ora ci apprestiamo a descrivere.

## 6.2 Somiglianza tra segnali: stato dell'arte

Obiettivo primario in questa fase è quello di trovare un metodo sistematico per caratterizzare la somiglianza di più segnali. La lista degli approcci in grado di quantificare se due o più segnali hanno un andamento simile è molto vasta e un suo elenco approfondito esula dallo scopo di questa tesi. Ne prenderemo in considerazione, quindi, un sottoinsieme ristretto. Iglesias et al. [39] prendono in considerazione una serie di misure di somiglianza basate sulla distanza tra vettori (i.e. segnali in questa trattazione). La prima misura della distanza considerata è la classica distanza euclidea, che per i vettori:

$$\begin{aligned}\vec{x} &= (x_1, x_2, \dots, x_n) \in \mathbb{R}^n \\ \vec{y} &= (y_1, y_2, \dots, y_n) \in \mathbb{R}^n\end{aligned}\tag{6.1}$$

è definita come:

$$d_{euclidean}(\vec{x}, \vec{y}) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_n - y_n)^2}\tag{6.2}$$

La Mahalanobis distance rispetto alla precedente tiene conto anche della correlazione dei segnali tramite la matrice di covarianza  $S$ :

$$d_{mahalanobis}(\vec{x}, \vec{y}) = \sqrt{(\vec{x} - \vec{y})^T S^{-1} (\vec{x} - \vec{y})}\tag{6.3}$$

Altra misura proposta è l'indice di correlazione di Pearson definito come:

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}\tag{6.4}$$

dove  $\bar{x}$  e  $\bar{y}$  sono la media di  $x$  e di  $y$ , rispettivamente.

Si termina con il Dynamic Time Warping (DTW), un algoritmo che permette di trovare una corrispondenza ottima tra due sequenze, attraverso una distorsione non lineare rispetto alla variabile indipendente (tipicamente il tempo). Quindi, indirettamente, misura la somiglianza tra due sequenze temporali che possono variare in velocità.

Lhermitte et al. [37] propongono le stesse misure appena descritte aggiungendo alla lista la distanza di Manhattan così definita:

$$d_{manhattan}(\vec{x}, \vec{y}) = \sum_{i=1}^n |p_i - q_i| \quad (6.5)$$

Si passa quindi a misure basate su PCA (vedi *sottosezione 5.3.1*) che rimappano i segnali in uno spazio multidimensionale diverso da quello originale, per mettere in evidenza la varianza dei dati (vedi *sottosezione 5.3.1*). Un'altra possibilità considerata è quella di confrontare i segnali nel dominio della frequenza usando la trasformata di Fourier e analizzandone i coefficienti, oppure studiare il segnale di correlazione incrociata tra segnali.

Serrà et al. [38] rimarca quanto detto nei precedenti articoli citati e conclude la lista introducendo l'utilizzo di modelli di autoregressione. L'idea è quella di costruire un modello dei due segnali presi in considerazione tramite uno dei numerosi metodi di autoregressione disponibili (e.g. AR, ARMA, ARIMA, ARIMAX) e di testare la loro somiglianza basandosi sui parametri restituiti dal modello stesso.

Per questa tesi è stato scelto di utilizzare un metodo basato sulla correlazione incrociata dei segnali, che nello specifico prende in considerazione la loro covarianza incrociata o Cross Covariance (xcov). La scelta è ricaduta su questo metodo per la semplicità di implementazione (è facile trovarlo in molte librerie e programmi), per la velocità di esecuzione e la facilità di interpretazione. Per ulteriori dettagli si veda la *sezione 6.3*.

## 6.3 Cross-covariance nell'analisi dei segnali

La covarianza, com'è facile intuire dal nome, indica la tendenza di due variabili a co-variare, quindi a “muoversi”, assieme. Se  $cov(X, Y)$  è positiva significa che grandi valori di  $X$  sono associati a grandi valori di  $Y$  o, in maniera analoga, a piccoli valori di  $X$  sono associati a piccoli valori di  $Y$ . Se invece la covarianza è negativa vale il contrario di quanto appena detto: piccoli valori della variabile  $X$  sono associati a grandi valori della variabile  $Y$  e viceversa.

La covarianza è una misura che dipende dalla scala usata per i valori coinvolti, per questo viene spesso normalizzata, dividendola per  $\sigma_x \cdot \sigma_y$ , ottenendo così la *correlazione*, che ha i vantaggi di essere priva di unità di misura e di variare nell'intervallo  $[-1; 1]$ . Questo la rende una misura facilmente gestibile, che descrive la relazione lineare tra due variabili.

Per l'utilizzo nell'elaborato, tuttavia, è stata usata una funzione strettamente collegata alla covarianza: la *covarianza incrociata* o *cross-covariance*. Essa misura la somiglianza tra  $x$  e la copia traslata nel tempo di  $y$ , come funzione del tempo di traslazione. Nella teoria dei segnali la covarianza incrociata, spesso confusa con la correlazione incrociata, è utilizzata come misura di somiglianza tra due segnali. Questa sua proprietà viene comunemente sfruttata per trovare delle particolari *feature* in un segnale sconosciuto a priori, con possibili applicazioni nei problemi di *pattern recognition* e nell'analisi crittografica. Rimarchiamo che la cross-covariance è una funzione del tempo relativo tra due segnali e, come tale, viene anche chiamata *sliding dot product*.

La funzione Matlab utilizzata per il calcolo della covarianza incrociata è  $xcov(x, y)$  e restituisce la covarianza incrociata tra due sequenze tempo-discrete  $x$  e  $y$ . Essa calcola la media degli input, quindi la sottrae dagli ingressi stessi e termina con una chiamata alla funzione  $xcorr$  (*cross-correlation*). Se  $x$  e  $y$  hanno lunghezze diverse, la funzione esegue un *padding* sul segnale più corto in modo da uguagliare la lunghezza di quello più lungo.  $xcov$  non esegue alcun tipo di controllo di errore se non quello sul numero degli ingressi delegando questo compito alla funzione  $xcorr$ .

La vera covarianza incrociata di due processi aleatori stazionari  $x_n$  e  $y_n$  corrisponde quindi alla correlazione incrociata delle due sequenze dopo che la loro

media è stata rimossa:

$$\phi_{x,y}(m) = E\{(x_{n+m} - \mu_x)((y_n - \mu_y)^*)\} \quad (6.6)$$

dove  $\mu_x$  e  $\mu_y$  sono le medie dei due processi aleatori stazionari, l'asterisco denota l'operazione di complesso coniugato ed  $E$  è l'operatore di valore atteso. *xcov* può solo stimare la sequenza risultante perchè, in pratica, è disponibile solo una porzione finita di una delle possibili realizzazioni dell'intero processo aleatorio il quale ha lunghezza infinita [1][2].

Di default, *xcov* calcola la covarianza senza normalizzazione:

$$c_{xy}(m) = \begin{cases} \sum_{n=0}^{N-m-1} \left( x_{n+m} - \frac{1}{N} \sum_{i=0}^{N-1} x_i \right) \left( y_n^* - \frac{1}{N} \sum_{i=0}^{N-1} y_i^* \right) & \text{if } m \geq 0 \\ c_{yx}^*(-m) & \text{if } m < 0 \end{cases} \quad (6.7)$$

Il vettore in output,  $c$ , ha elementi dati da:

$$c(m) = c_{xy}(m - N), \quad m = 1, \dots, 2N - 1 \quad (6.8)$$

La funzione covarianza richiede la normalizzazione per eseguire una stima corretta, che è possibile controllare usando in ingresso il parametro opzionale *scaleopt*.

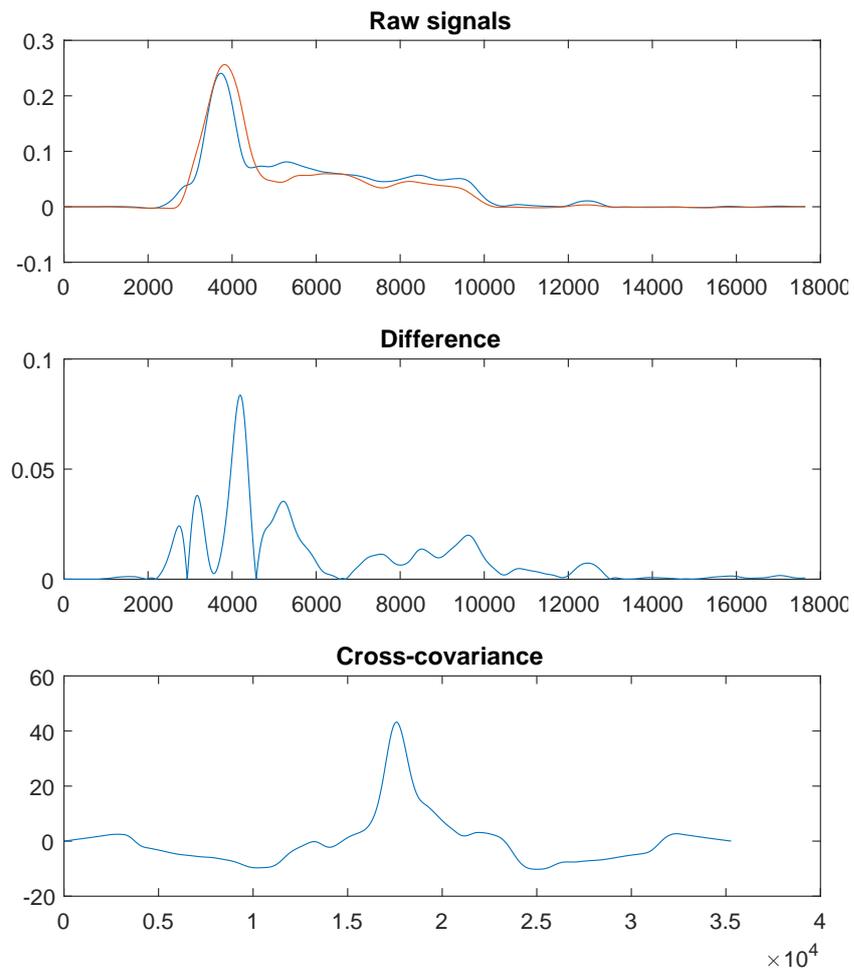
## 6.4 Metodo di selezione sistematico

I tentativi di creare un metodo di selezione dei segnali di trial basato sulla loro somiglianza è stata l'unione, in diverse combinazioni, di due analisi distinte:

1. Calcolo della **cross-covariance** (**covarianza incrociata**) tra i vari segnali di trial. Questa funzione, basata sulla *cross-correlation*, viene utilizzata nella teoria dei segnali per quantificare la somiglianza tra 2 funzioni (vedi *sezione 6.3*).

2. **Differenza** tra 2 segnali di trial una volta allineati: si basa sulla semplice osservazione che 2 segnali simili, se adeguatamente allineati, hanno come differenza un segnale prossimo allo zero.

Queste due misure di bontà sono state opportunamente inserite in un flusso di operazioni che forma l'algoritmo di selezione, descritto nella *sottosezione 6.4.1*.



### 6.4.1 Algoritmo di selezione

Si consideri il soggetto  $S_n$  e il movimento  $M_h$ . Data la coppia  $(S_n, M_h)$  ne vengono estratti i 6 trial relativi  $T_1, \dots, T_6$ , sui quali viene applicato il pre-processing con *smoothing*, *normalizzazione* e *zero-signal*.

1. Per ogni possibile coppia  $(T_i, T_j)$  calcolo la differenza  $T_{sub}^{ij} = T_i - T_j$ . Calcolo successivamente  $diff_{ij} = \max(T_{sub}^{ij}) - \min(T_{sub}^{ij})$  che intuitivamente rappresenta lo spessore delle soglie che limitano la funzione  $T_{sub}^{ij}$ . Quanto più questa differenza è bassa tanto più i segnali che la generano sono simili e quindi significativi.
2. Per ogni possibile coppia  $(T_i, T_j)$  calcolo la funzione *cross-covariance*,  $T_{xcov}^{ij}$ , tra  $T_i$  e  $T_j$ . Calcolo successivamente il picco del segnale  $T_{xcov}^{ij}$  chiamato  $\max(T_{xcov}^{ij})$ : maggiore è il picco, più buono sarà il segnale per i nostri scopi. Viene sfruttata la capacità della funzione cross-covariance di riconoscere la somiglianza tra due segnali.
3. A questo punto otteniamo 2 misure,  $\max(T_{xcov}^{ij})$  e  $diff_{ij}$ , per ognuna delle  $\binom{num_{trial}}{2} = \binom{6}{2} = 15$  coppie  $(T_i, T_j)$ . Ora per ogni canale sommiamo, separatamente, i valori di queste misure, ottenendo, per il canale  $c$ -esimo:

$$\Sigma_c^{diff} = \sum_{i=1}^6 \sum_{j=1}^6 diff_{ij} \quad (6.9)$$

$$\Sigma_c^{xcov} = \sum_{i=1}^6 \sum_{j=1}^6 \max(T_{xcov}^{ij}) \quad (6.10)$$

Il fatto di aver preso in considerazione, per ogni canale, ogni coppia di trial possibile, permette di non trascurare alcuna somiglianza e differenza, usando la totalità dei dati che si ha a disposizione.

4. Il prossimo passo è quello di ordinare i canali secondo i due valori  $\Sigma_c^{diff}$  e  $\Sigma_c^{xcov}$ , separatamente, e assegnare un punteggio progressivo in base alla loro posizione dopo l'ordinamento, partendo da punteggio 12 per il canale più promettente fino ad arrivare a punteggio 1 per il canale meno promettente. Così facendo il canale  $c$ -esimo avrà associati i 2 punteggi  $Score_c^{diff} \in [1, 12]$  e  $Score_c^{xcov} \in [1, 12]$ .

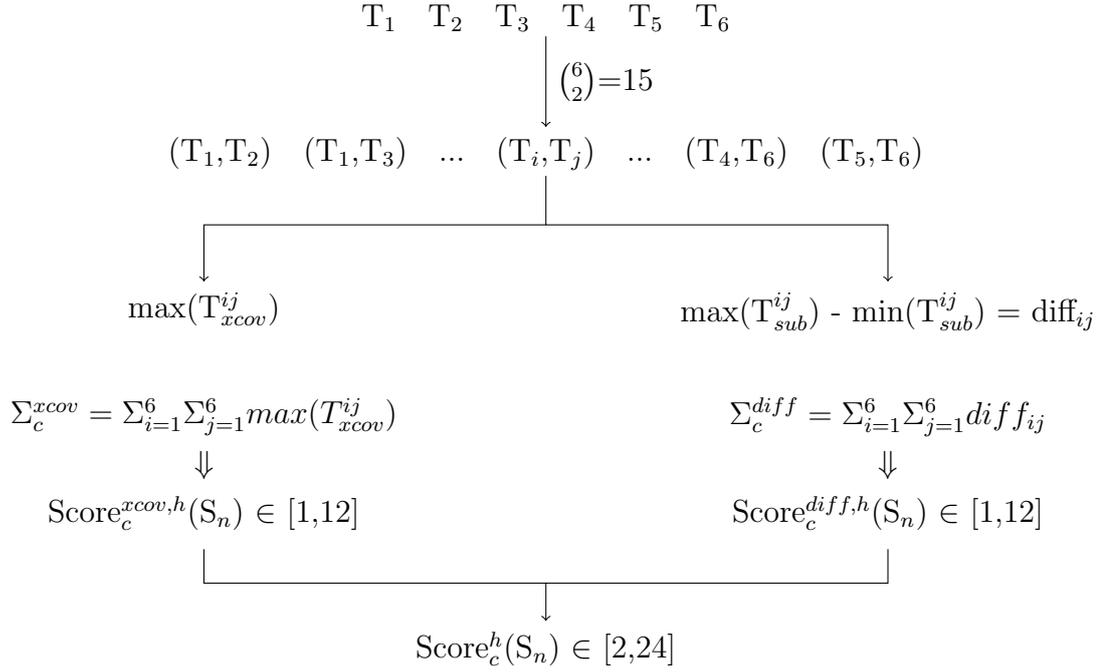


Figura 6.4: Movimento  $M_h$ , Soggetto  $S_n$ , Canale  $c$

5. Per ogni canale sommiamo i punteggi parziali  $Score_c^{diff, h}$  e  $Score_c^{xcov, h}$ , ottenendo il punteggio totale  $Score_c^h$  che inserisco in una matrice che incrocia, per ogni movimento, canali e soggetti (*vedi Tabella 6.2*).
6. L'ultimo passo è quello di sommare i punteggi parziali di ogni canale su tutti i soggetti ottenendo così un risultato associato al singolo canale che spazia su tutto lo spazio dei soggetti. Avvalendosi di ciò è possibile stilare una classifica dei canali per un determinato movimento.

$$Score_c^h = \sum_{\forall S_n \in \{subjects\}} Score_c^h(S_n) \quad \begin{array}{l} \forall M_h \in \{movements\} \\ \forall c \in \{channels\} \end{array} \quad (6.11)$$

M1	ch1	ch2	ch3	ch4	ch5	ch6	ch7	ch8	ch9	ch10	ch11	ch12
S1	21	23	16	10	16	10	16	3	10	14	9	8
S2	24	5	17	9	14	11	11	19	16	22	3	5
S3	15	17	22	17	5	6	15	18	9	22	6	4
S4	12	19	13	8	13	13	20	20	7	24	2	5
S5	22	15	9	4	9	12	20	24	9	16	2	14
S6	16	23	23	10	8	13	17	13	5	20	4	4
S7	22	24	18	5	6	11	9	17	18	14	5	7
S8	13	19	13	15	22	17	4	23	11	8	8	3
S9	11	19	18	2	11	16	17	20	10	6	22	4
S10	18	23	15	5	13	8	10	23	20	11	2	8
S11	24	21	17	17	13	14	9	8	7	20	3	3
S12	20	21	17	13	14	15	15	13	12	9	4	3
S13	22	23	17	12	13	7	12	19	18	7	3	3
S14	23	23	20	13	8	16	13	16	7	11	3	3
S15	21	21	19	13	17	9	8	22	13	7	3	3
S16	23	17	14	16	11	3	7	23	14	16	8	4
S17	21	22	15	17	8	9	7	13	16	22	4	2
S18	18	23	20	19	13	8	10	17	16	6	4	2
S19	18	19	15	9	16	18	6	13	17	19	3	3
S20	7	21	24	12	10	6	7	21	18	16	8	6
$\Sigma$ ch	371	398	342	226	240	222	233	345	253	290	106	94

Tabella 6.2: Matrice dei punteggi totali  $Score_c^h$  del movimento M1 per un campione di 20 soggetti. Si noti come il feedback visuale dato dai colori indichi, in questo esempio, la bontà dei canali 1, 2 e 8. Nella fattispecie l'alta concentrazione di verde in una colonna suggerisce un buon punteggio per il canale relativo.

### 6.4.2 Interpretazione e commento dei risultati

Le immagini in *Tabella 6.2* e in *Figura 6.5* accreditano i risultati esposti nella *Tabella 6.1* e nella *Figura 6.3*: la tendenza evidenziata dal metodo appena descritto ricalca molto fedelmente i risultati ottenuti con il metodo empirico sugli stessi 20 soggetti. A riprova di ciò, i primi tre canali selezionati con punteggio più alto coincidono: essi sono il canale 1, il 2 e l'8.

## 6.5 Verifica della correttezza

Abbiamo deciso di verificare la correttezza del metodo di selezione dei canali eseguendo il procedimento inverso rispetto a quello descritto nel precedente *Capitolo 6*: si parte dai risultati dell'algoritmo applicato ad un ad un movimento qualunque (e.g. movimento M2 nell'esempio seguente) per poi confrontarli con quelli ottenuti da un'osservazione empirica-qualitativa. Ricordiamo che quest'ultimi sono frutto di un'osservazione individuale che ha lo scopo di elencare,

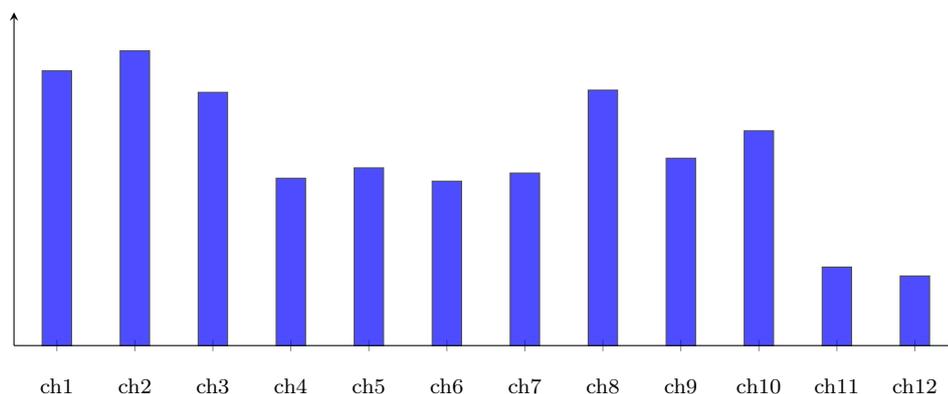


Figura 6.5: Istogramma rappresentativo della matrice in *Tabella 6.2*. Sull’asse delle ordinate sono rappresentate, per ogni canale, le somme dei punteggi per tutti i soggetti:  $\sum_{h=S1}^{S20} Score_c^h$ .

per ognuno dei 40 soggetti, una volta fissato il movimento, quali sono i 3 canali che mostrano l’andamento più simile tra i propri 6 trial. Pertanto ci si aspetta di ottenere un risultato quanto più simile possibile a quello ottenuto dall’algoritmo, in maniera tale da provarne la correttezza. Al contempo non si esclude la presenza di errori di valutazione dovuti ad un’assegnazione dei punteggi soggettiva.

S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14
3	4	3	3	2	3	2	4	6	2	3	3	3	3
5	7	8	9	7	7	3	5	7	3	8	7	7	7
6	8	9	10	8	10	4	6	9	9	9	8	9	8
S15	S16	S17	S18	S19	S20	S21	S22	S23	S24	S25	S26	S27	
3	2	7	3	3	2	3	2	6	3	3	3	3	
4	3	8	8	4	3	7	5	7	4	4	7	7	
9	9	10	10	10	9	8	7	10	9	10	10	9	
S28	S29	S30	S31	S32	S33	S34	S35	S36	S37	S38	S39	S40	
4	3	5	2	3	7	3	7	4	2	3	7	6	
7	9	8	7	4	9	7	8	7	6	8	9	7	
10	10	9	8	10	10	9	10	10	10	9	10	8	

Tabella 6.3: Risultato della selezione dei 3 canali più significativi basandosi su un’osservazione qualitativa per il movimento M2 su 40 soggetti.

Si confrontino ora gli istogrammi riassuntivi nella *Figura 6.6* (risultati empirici-sperimentali) e nella *Figura 6.7* (risultati dell’algoritmo). Si può notare un andamento simile tra i due grafici, ad esempio considerando i migliori 6 canali si ottiene:

- *osservazione empirica*: canali 3, 7, 9, 10, 8, 4
- *algoritmo*: canali 3, 7, 9, 8, 10, 4

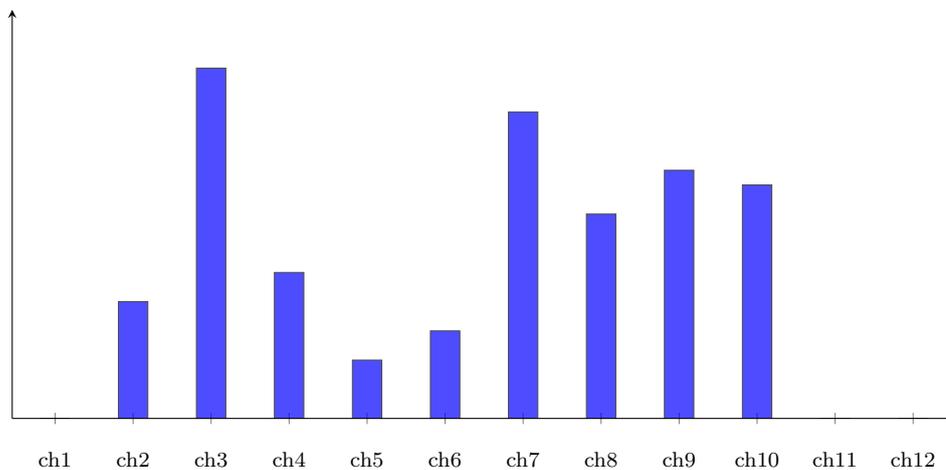


Figura 6.6: Istogramma rappresentativo della *Tabella 6.3*.

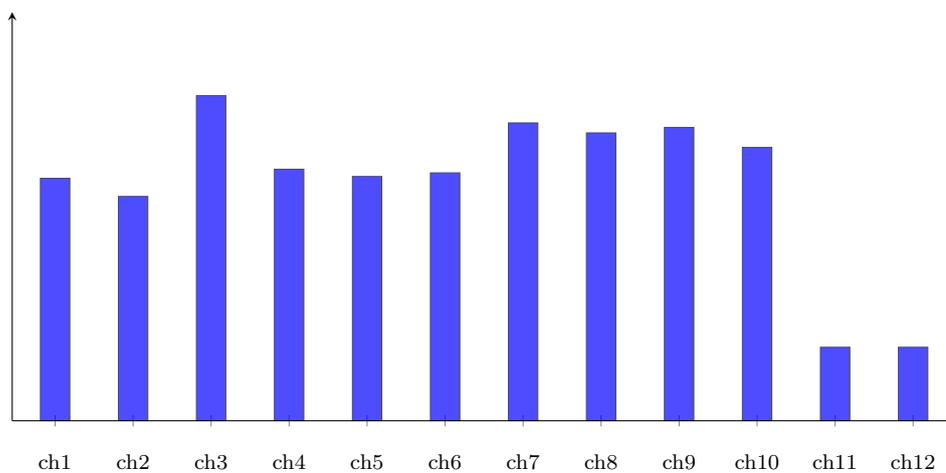


Figura 6.7: Istogramma rappresentativo dei risultati dell'algoritmo per il movimento M2 su 40 soggetti.

6 canali su 6 sono uguali, questo ci suggerisce che, con buona approssimazione, quello che è stato catturato dall'osservatore viene rilevato e riproposto anche dall'algoritmo.



# Capitolo 7

## Analisi dell'algoritmo

Viene presa in considerazione la possibilità di affinare l'algoritmo con variazioni sul tema base esposto nelle sezioni precedenti. In una trattazione che vede così tante variabili in gioco risulta essenziale capire come migliorino, peggiorino o semplicemente cambino i risultati al mutare di questi input.

### 7.1 Combinazione delle elaborazioni

È stata verificata la dipendenza del metodo dal tipo di elaborazione che i segnali ricevevano. Rimarchiamo i 3 step coinvolti nell'elaborazione dei segnali sEMG grezzi dopo la fase di DWT:

1. Smoothing
2. Normalizzazione
3. Traslazione

Nella fattispecie, la dipendenza è stata testata rieseguendo l'algoritmo di selezione descritto nella *sottosezione 6.4.1* per per tutte le possibili combinazioni dei punti 1, 2 e 3 e osservando eventuali variazioni dei risultati rispetto al caso base che comprende tutti e tre i punti. Vengono elencate di seguito le combinazioni provate:

- RAW: segnale grezzo senza alcuna elaborazione se non la DWT
- Solo smoothing
- Solo normalizzazione
- Solo traslazione

- Smoothing e normalizzazione
- Smoothing e traslazione
- Normalizzazione e traslazione
- Elaborazione completa: smoothing, normalizzazione ed traslazione

Dalle tabelle di pagina seguente si può constatare che esiste una forte dipendenza dell’algoritmo dalla normalizzazione. Questa è un presupposto essenziale in quanto il risultato della differenza tra segnali dipende fortemente dal fatto che questi siano distribuiti in intervalli simili o diversi e non si potrà avere uniformità nella valutazione a meno di non riportare tutti i segnali nello stesso intervallo di valori per le ordinate, vale a dire, normalizzandoli. Era facile intuire che tutto dipendesse dalla normalizzazione in quanto si sta cercando di ”misurare” la somiglianza di segnali.

M1	ch1	ch2	ch3	ch4	ch5	ch6	ch7	ch8	ch9	ch10	ch11	ch12
S1	13	15	13	14	13	13	14	13	11	13	13	11
S2	13	13	14	13	13	13	13	13	12	13	13	13
S3	13	14	13	11	14	14	14	13	11	13	13	13
S4	10	15	14	13	14	14	11	12	14	15	12	12
S5	15	13	13	12	13	14	13	13	11	13	13	13
S6	12	11	14	15	12	14	13	13	13	13	13	13
S7	13	13	13	13	13	13	13	13	12	13	13	14
S8	12	13	14	12	14	14	13	15	14	11	13	11
S9	14	12	13	13	13	13	13	13	13	13	13	13
S10	13	14	14	12	13	15	14	14	12	10	13	12
S11	13	12	11	12	13	13	13	15	13	15	13	13
S12	13	13	13	14	14	12	14	13	13	12	13	12
S13	13	13	14	10	13	14	15	13	13	13	12	13
S14	15	15	13	13	13	13	11	13	11	13	13	13
S15	14	12	14	13	14	12	11	15	14	11	12	14
S16	13	16	13	13	13	13	14	12	13	10	13	13
S17	13	12	13	12	13	13	14	15	12	16	10	13
S18	14	13	14	13	13	14	13	12	12	13	12	13
S19	11	12	12	12	12	13	13	14	17	14	13	13
S20	13	16	13	11	13	13	13	13	13	12	14	12
Σch	260	267	265	251	263	267	262	267	254	256	254	254

Tabella 7.1: Matrice dei punteggi totali  $Score_c^h$  del movimento M3 per un campione di 20 soggetti con elaborazione composta solo da smoothing e traslazione e senza normalizzazione. Si noti come il feedback visivo dato dai colori indichi, in questo esempio, una scarsa variabilità tra i risultati fornendo pochissima informazione per quanto riguarda la scelta dei canali.

## 7.2 Metodo di valutazione pesato

Un'altra possibilità esplorata nello svolgimento dell'esperienza è stata quella di associare pesi diversi ai punteggi di covarianza incrociata e differenza. Nella fattispecie, per ogni movimento  $M_h$  preso in considerazione:

- Il punteggio del canale  $c$ -esimo associato alla covarianza incrociata  $Score_c^{diff,h}$  viene pesato con il fattore  $w_{diff}$
- Il punteggio del canale  $c$ -esimo associato alla covarianza incrociata  $Score_c^{xcov,h}$  viene pesato con il fattore  $w_{xcov}$

A differenza del metodo precedente descritto nella *sottosezione 6.4.1*, sono stati sommati i punteggi parziali per covarianza e differenza con somme separate:

$$Score_c^{diff,h} = \sum_{\forall S_n \in \{subjects\}} Score_c^{diff,h}(S_n) \quad \begin{array}{l} \forall M_h \in \{movements\} \\ \forall c \in \{channels\} \end{array} \quad (7.1)$$

$$Score_c^{xcov,h} = \sum_{\forall S_n \in \{subjects\}} Score_c^{xcov,h}(S_n) \quad \begin{array}{l} \forall M_h \in \{movements\} \\ \forall c \in \{channels\} \end{array} \quad (7.2)$$

Ottenendo un punteggio per canale dato da:

$$Score_c^h = w_{xcov} \cdot Score_c^{xcov,h} + w_{diff} \cdot Score_c^{diff,h} \quad (7.3)$$

Il peso è stato distribuito in modo che  $w_{xcov} + w_{diff} = 1$ .

Quello che si può notare al variare del peso è che, facendo predominare il punteggio dovuto alla covarianza incrociata rispetto a quello dato dalla differenza, si ottiene un aumento del divario tra i canali migliori e quelli giudicati meno appetibili. Viceversa, l'aumento del fattore  $w_{diff}$  genera un effetto livellante, appiattendole le differenze di punteggio tra i vari canali. Globalmente invece, nella maggior parte dei casi, viene mantenuta la gerarchia ottenuta con il peso paritario  $w_{xcov} = w_{diff} = 0.5$ .

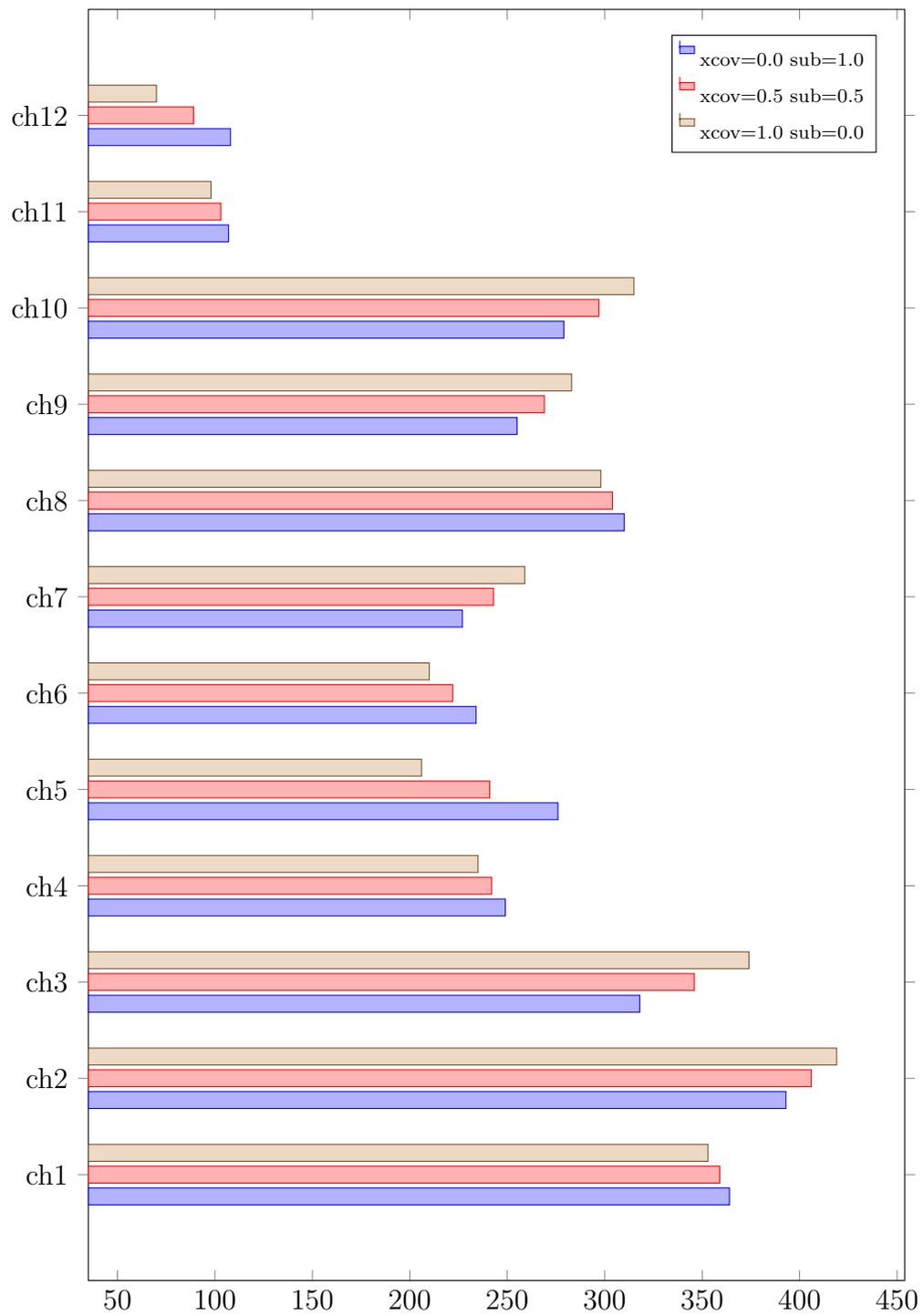


Figura 7.1: Punteggi assegnati dall'algorithm ad ogni canale per 3 diverse configurazioni pesate dei punteggi di covarianza incrociata e differenza - Movimento M1 - normalizzazione per segnale con trial non allineati.

La *Figura 7.1* mostra, in un esempio per il movimento M1, quali sono i canali più suscettibili alla variazione di peso nel punteggio di *xcov* e *diff*. Le 3 coconfigurazioni prese in considerazione riflettono ad esempio che i canali 1 e 11 risentono in maniera marginale del peso sbilanciato su uno dei due punteggi *diff* e *xcov*, al contrario dei canali 3, 5, 10 e 12 che mostrano una differenza sensibile del punteggio al variare di questi pesi.

### 7.3 Normalizzazione sul canale

La dipendenza dell'algoritmo dalla normalizzazione descritta nella *sezione 7.1* ci ha spinto ad indagare in questa direzione, approfondendo metodi alternativi per l'elaborazione dei segnali.

La normalizzazione di ogni singolo segnale costituisce un aiuto essenziale per evidenziarne le peculiarità dell'andamento rispetto ad altri segnali. Riporta inoltre il codominio ad un intervallo unificato e quindi facilmente trattabile. Al contempo però, non tiene conto del fatto che anche l'evoluzione relativa tra due segnali, che ad esempio possono avere una forma simile ma su scala diversa, costituisce di per sè una fonte di informazione. Si è scelto quindi di ispezionare la possibilità di normalizzare i segnali usando i valori massimo e minimo di ogni canale. Questo tipo di normalizzazione che chiameremo *per canale*, invece di normalizzare ogni segnale (trial) nella maniera più classica secondo il suo massimo e minimo eseguendo:

$$signal_{normalized} = \frac{signal - \min(signal)}{\max(signal) - \min(signal)}; \quad (7.4)$$

esegue il ridimensionamento nella maniera seguente:

$$signal_{normalized}^{ch} = \frac{signal - \min(channel)}{\max(channel) - \min(channel)}; \quad (7.5)$$

I risultati di questa prova sono illustrati in *Figura 7.2*, *Figura 7.3* e *Figura 7.4*.

Dall'istogramma della *Figura 7.2* si vede facilmente come l'introduzione della normalizzazione secondo il canale abbia introdotto un'inversione di tendenza piuttosto marcata per i canali 4, 5, 6, 7, 10, 11 e 12. Si noti che l'istogramma

rappresenta solamente i punteggi di  $x_{cov}$ , i quali risentono limitatamente della nuova normalizzazione. La funzione covarianza incrociata infatti, è sensibile, seppur marginalmente, alla variazione in ampiezza dei segnali coinvolti.

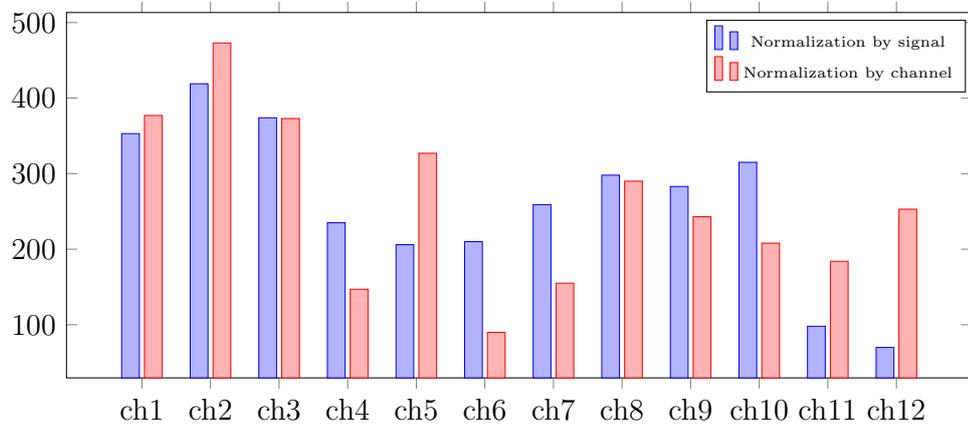


Figura 7.2: Normalizzazione: per canale VS per segnale - M1, 40 Soggetti, Segnali non allineati -  $w_{xcov} = 1.0$ ,  $w_{sub} = 0.0$

L'istogramma in *Figura 7.3* mostra solo i punteggi *diff*. Qui la normalizzazione sul canale ribalta completamente i risultati ottenuti con la normalizzazione per segnale.

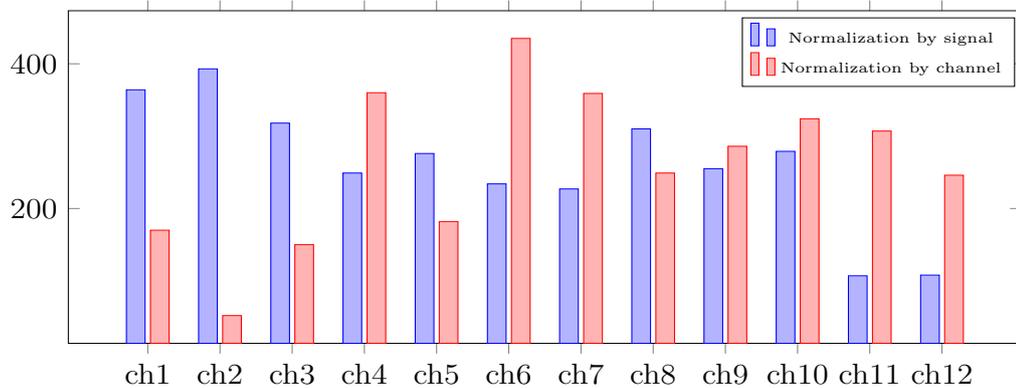


Figura 7.3: Normalizzazione: per canale VS per segnale - M1, 40 Soggetti, Segnali non allineati -  $w_{xcov} = 0.0$ ,  $w_{sub} = 1.0$

Il fatto che quest'ultimi non vengano riconfermati dai nuovi punteggi non stupisce: la differenza tra due segnali usata come misura per la somiglianza del loro

andamento ha senso solo nel caso in cui questi siano normalizzati singolarmente, annullando la grossa differenza che si avrebbe tra due segnali molto simili nell'andamento, ma con valori di picco molto differenti.

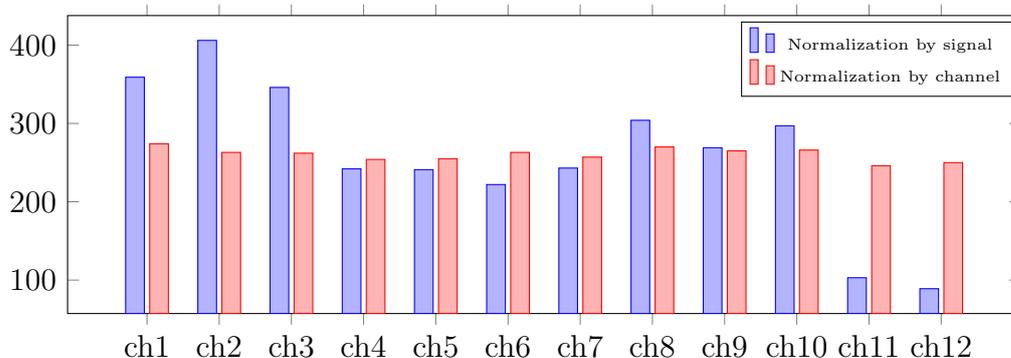


Figura 7.4: Normalizzazione: per canale VS per segnale - M1, 40 Soggetti, Segnali non allineati -  $w_{xcov} = 0.5$ ,  $w_{sub} = 0.5$

L'effetto della normalizzazione per canale è palesato nel grafico a barre rosse della *Figura 7.4*, dove i punteggi di  $xcov$  e  $diff$  sono pesati in maniera uguale. Qui i risultati delle due parti si compensano a vicenda dando luogo ad un istogramma livellato e praticamente piatto, che non evidenzia una classifica ben marcata di quelli che potrebbero essere i canali migliori e quelli peggiori. Al contrario, l'istogramma a barre blu nella stessa immagine che mostra come la normalizzazione classica sia in grado di evidenziare alti e bassi di ogni canale.

Alla luce di questi risultati possiamo affermare che nel caso in cui si utilizzi una normalizzazione per canale si considera inutile o addirittura deleterio per la correttezza degli stessi l'uso di un peso per il punteggio di *differenza*  $w_{sub}$  diverso da zero. Ciò fa sì che in questo frangente il punteggio relativo alla covarianza incrociata acquisisca totale decisione con  $w_{xcov} = 1.0$ .

Diamo un riscontro visivo che motivi gli scarsi risultati attribuiti dall'algoritmo ad alcuni canali. Prendiamo per esempio il canale 11: come si nota dalla *Figura 7.5* e dalla *Figura 7.6*, nella normalizzazione per segnale la sovrapposizione dei trial si tramuta in una sorta di rumore attraverso il quale non è intuibile, nemmeno qualitativamente, alcuna tendenza alla ripetizione di un certo pattern. È proprio questo ciò che rileva l'algoritmo assegnando scarsi risultati a questo canale.

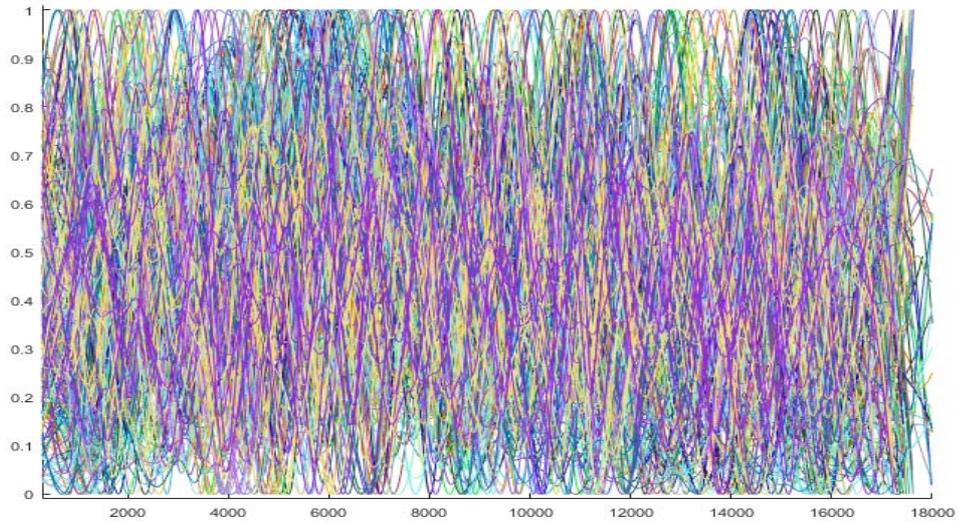


Figura 7.5: 6 trial di tutti e 40 i soggetti, canale 11, normalizzati per segnale.

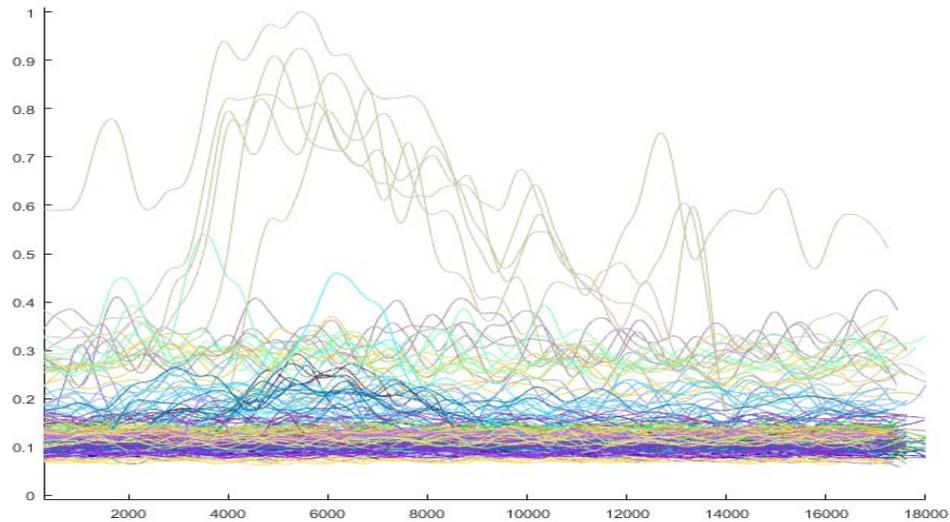


Figura 7.6: 6 trial di tutti e 40 i soggetti, canale 11, normalizzati per canale.

Viceversa, come mostrano bene la *Figura 7.7* e la *Figura 7.8*, il canale 2 mostra una certa ripetibilità nei pattern di ogni soggetto, sia nel caso in cui il trial sia normalizzato per segnale sia in quello che sia normalizzato per canale sul singolo movimento.

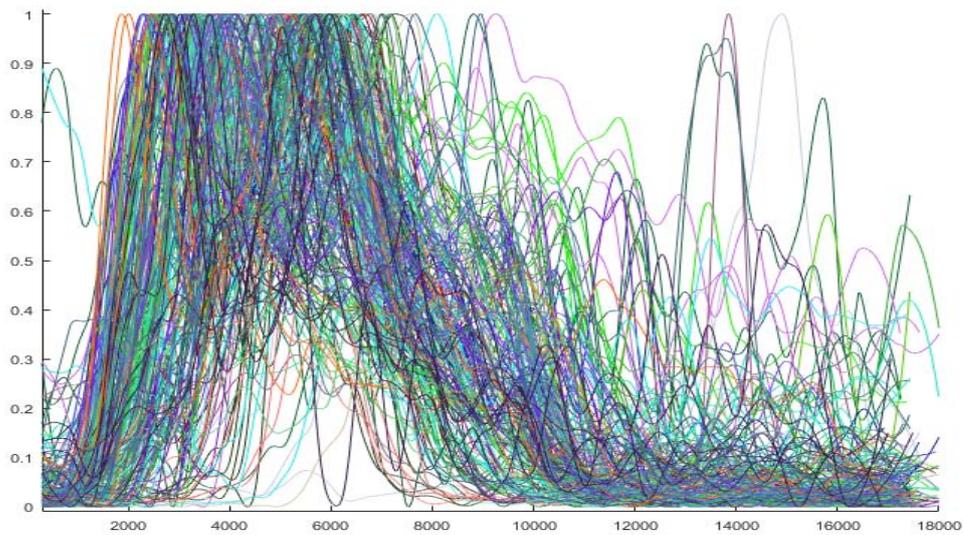


Figura 7.7: 6 trial di tutti e 40 i soggetti, canale 2, normalizzati per segnale.

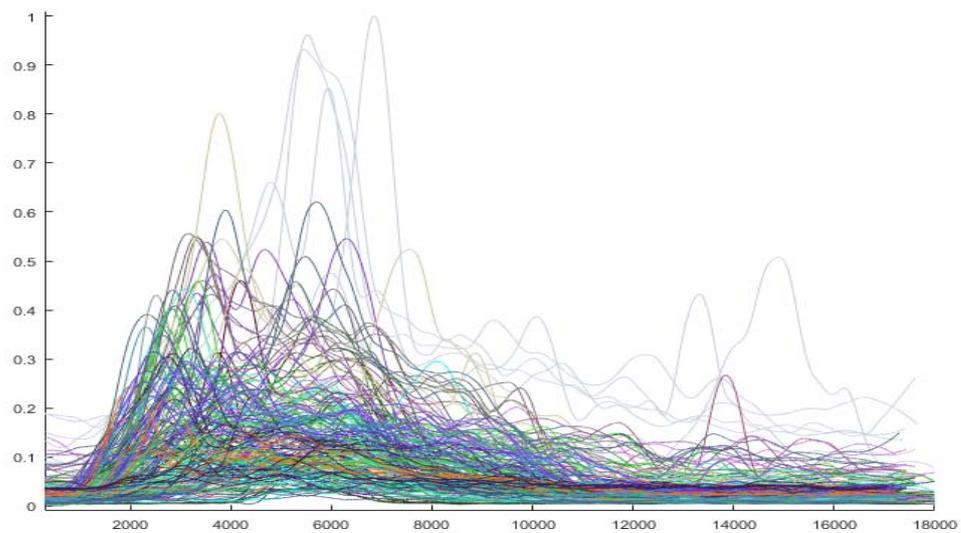


Figura 7.8: 6 trial di tutti e 40 i soggetti, canale 2, normalizzati per canale.

Possiamo facilmente evidenziare quali sono i canali che vengono maggiormente rivalutati calcolando la differenza tra i punteggi ottenuti con i due diversi tipi di normalizzazione nel caso più significativo  $w_{xcov} = 1.0$  e  $w_{sub} = 0.0$ , come si evince dalla *Figura 7.9*.

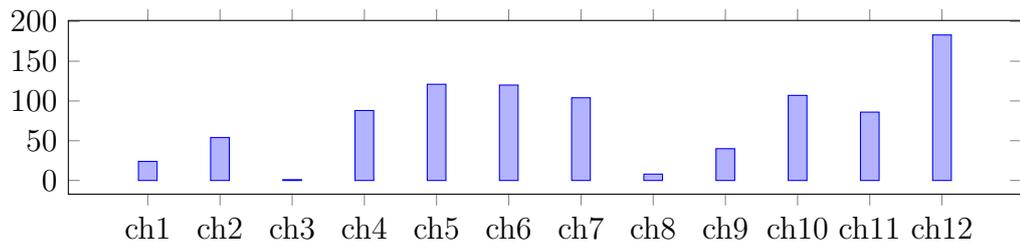
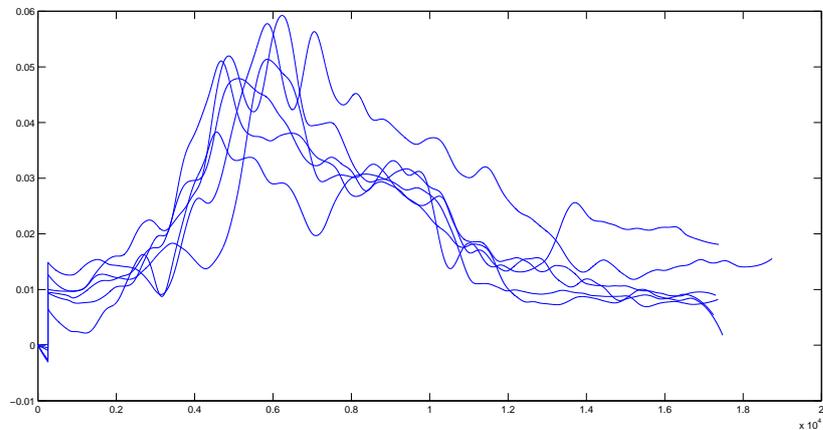


Figura 7.9: M1, 40 soggetti, segnali non allineati -  $w_{xcov} = 1.0$ ,  $w_{sub} = 0.0$

## 7.4 Allineamento dei segnali



I segnali di trial per ogni tripletta soggetto-movimento-canale sono stati allineati usando la funzione Matlab *alignsignals*. Non potendo allineare i 6 trial in una sola volta, si è deciso di allinearli a coppie usando di volta in volta come riferimento il primo trial di quell'insieme e allineando i rimanenti 5 di conseguenza. Il risultato di questo processo è illustrato in *Figura 7.10*. La funzione *alignsignals* basa il suo funzionamento sul concetto di correlazione incrociata che come descritto nella *sezione 6.3*, è strettamente collegato al riconoscimento della somiglianza di due segnali.

Come si può constatare dagli istogrammi in *Figura 7.11*, *Figura 7.12* e *Figura 7.13*, l'aggiunta di un preprocessing basato sull'allineamento dei segnali non cambia sostanzialmente i risultati finali ottenuti in precedenza, serve tuttavia ad affinarli, rendendo maggiormente equo e veritiero il confronto fatto tra i segnali.

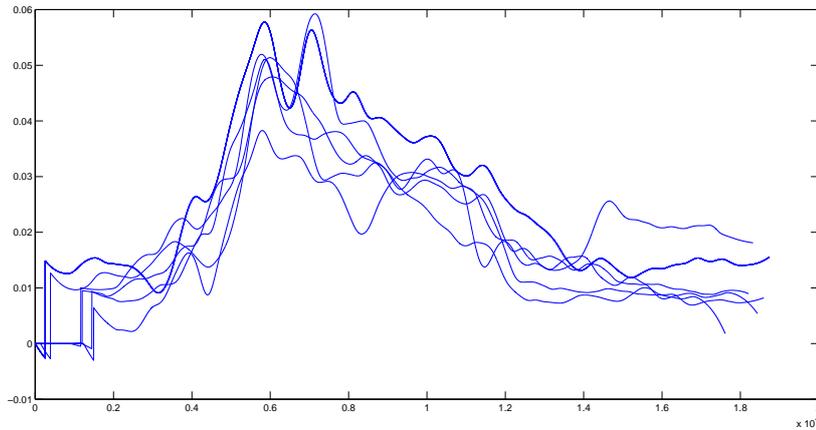


Figura 7.10: 6 trial del movimento M1 per il soggetto 1, Canale 1 non allineati e allineati, rispettivamente.

Questo accade in particolar modo nella differenza tra segnali in cui è più plausibile assegnare un punteggio alla sottrazione di segnali allineati che a quella di segnali non allineati. Il punteggio per la covarianza incrociata invece non subisce sostanziali cambiamenti se non quelli dovuti al troncamento dei segnali al fine di far rispettare i requisiti imposti da Matlab per eseguire la sottrazione.

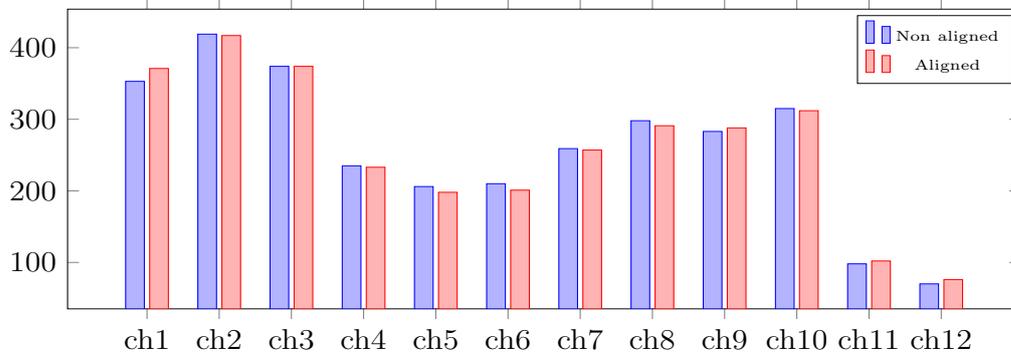


Figura 7.11: Segnali: allineati VS non allineati - M1, 40 Soggetti, normalizzazione per segnale -  $w_{xcov} = 1.0$ ,  $w_{sub} = 0.0$

Nonostante per gli istogrammi illustrati, i confronti effettuati e le conclusioni raggiunte in questa *sezione 7.3* fanno riferimento al movimento M1, esse possono essere estese in maniera più generale a movimenti diversi. Si è scelto di commentare esclusivamente il movimento M1 solo per semplicità di trattazione.

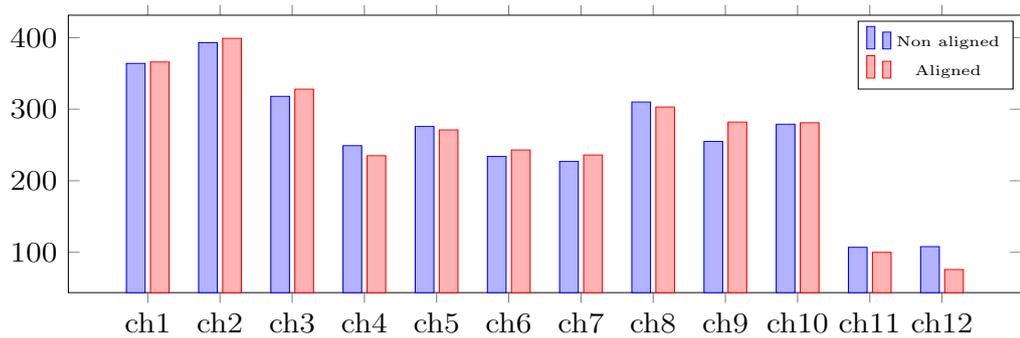


Figura 7.12: Segnali: allineati VS non allineati - M1, 40 Soggetti, normalizzazione per segnale -  $w_{xcov} = 0.0$ ,  $w_{sub} = 1.0$

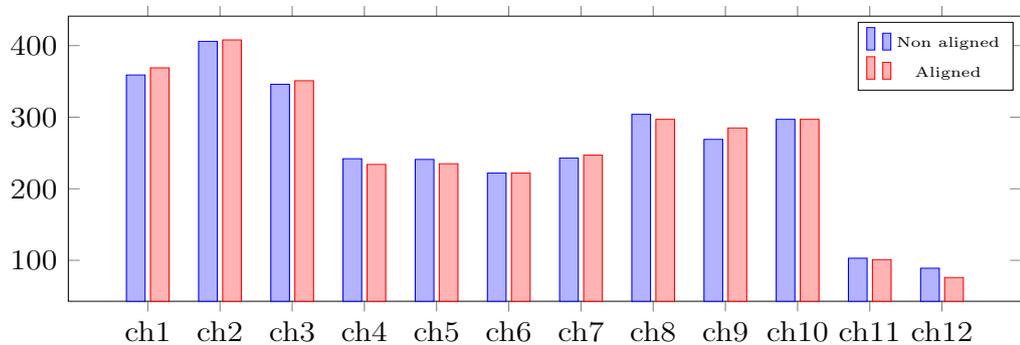


Figura 7.13: Segnali: allineati VS non allineati - M1, 40 Soggetti, normalizzazione per segnale -  $w_{xcov} = 0.5$ ,  $w_{sub} = 0.5$

## 7.5 Prestazioni

Le prove dei tempi sono state effettuate su un laptop Asus X550CL con le seguenti caratteristiche:

- Sistema operativo: Windows 8.1, 64 bit
- Processore: Intel(R) Core(TM) i7-3537U @ 2.00GHz
- Memoria RAM: 4 GB
- Software: Mathworks Matlab r2016b

impiegando circa 300 secondi per la selezione dei canali considerando 40 soggetti per un singolo movimento.

# Capitolo 8

## Risultati

### 8.1 Allenamento del modello GMM e regressione

Sulla base di quanto detto nella *sezione 1.3*, la matrice utilizzata per allenare il modello GMM, detta matrice di training, presenta sulle prime  $n$  colonne,  $n$  dei canali sEMG selezionati dall'algoritmo (i.e.  $\overline{ch_1}, \dots, \overline{ch_n}$ ) e sulle ultime  $m$  colonne, gli  $m$  giunti dei quali si intende stimare l'andamento a partire dai canali sEMG. In ogni riga invece sono presenti i dati dei primi 5 trial di tutti 40 i soggetti.

$$M_{training} = \begin{matrix} & \overline{ch_1} & \cdots & \overline{ch_n} & \overline{jnt_1} & \cdots & \overline{jnt_m} \\ \begin{matrix} S_1 \\ \vdots \\ S_1 \\ \vdots \\ S_{40} \\ \vdots \\ S_{40} \end{matrix} & \left( \begin{matrix} t_1 & \cdots & t_1 & t_1 & \cdots & t_1 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ t_5 & \cdots & t_5 & t_5 & \cdots & t_5 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ t_1 & \cdots & t_1 & t_1 & \cdots & t_1 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ t_5 & \cdots & t_5 & t_5 & \cdots & t_5 \end{matrix} \right) & \end{matrix} \quad (8.1)$$

Il singolo elemento della matrice  $t_p$  è un vettore colonna composto da  $D$  elementi, che rappresenta il  $p$ -esimo trial, una volta fissati il movimento, il soggetto e il canale:

$$t_p^T = [v_1, v_2, \dots, v_D] \in \mathfrak{R}^D, \quad p \in \{1, \dots, 6\} \quad (8.2)$$

In maniera analoga viene costruita anche la matrice di validazione, ovvero quella che conterrà i dati relativi ai trial sEMG sui quali stimare il movimento dei giunti per mezzo dell'algoritmo GMR. Nella matrice di validazione sono presenti solo i segnali EMG, mentre quelli dei giunti vengono stimati con una tecnica di regressione. I segnali EMG considerati sono quelli del sesto trial di ogni soggetto, non utilizzati nella costruzione del modello.

$$M_{validation} = \begin{matrix} & \overline{ch_1} & \cdots & \overline{ch_n} \\ S_1 & \left( \begin{matrix} t_6 & \cdots & t_6 \\ \vdots & \ddots & \vdots \\ S_{40} & t_6 & \cdots & t_6 \end{matrix} \right) \end{matrix} \quad (8.3)$$

## 8.2 Misure di accuratezza

Come descritto in precedenza, nella fase finale il GMR stima i segnali del *glove* (estratti dal database Ninapro) che rappresentano l'andamento dei vari giunti della mano. Avendo a che fare con segnali originali e segnali stimati è utile valutare la bontà della previsione generata dal modello attraverso una o più misure di accuratezza. Di seguito descriviamo le due misure utilizzate.

### Goodness Of Fit (GOF) basata su Normalized Mean Square Error (NMSE)

Questa prima misura di accuratezza viene calcolata a partire dal segnale del giunto di partenza, detto *original*, e dal segnale stimato dal modello, detto *estimate*, come indicato nell'Equazione 8.4:

$$GOF_{NMSE} = \frac{\|estimate - original\|^2}{\|estimate - mean(estimate)\|^2} \quad (8.4)$$

Il valore di  $GOF_{NMSE}$  varia tra  $-\infty$  che indica una cattiva stima e 1 che indica una stima perfetta. Allo scopo di valutare la bontà della predizione fatta dall'algoritmo GMR viene fatta la media degli indici GOF relativi ai diversi trial dei quali viene calcolata la stima tramite regressione:

$$accuracy_{NMSE} = \frac{1}{T} \sum_{i=0}^T GOF_{NMSE}(original_i, estimate_i) \in [-\infty, 1] \quad (8.5)$$

dove  $T$  è il numero di trial stimati dal modello, *original* è il segnale di *glove* come appare nel dataset Ninapro e *estimate* è il relativo segnale stimato.

### Coefficienti di correlazione

Il coefficiente di correlazione di due variabili casuali misura la loro dipendenza lineare. Se ciascuna delle due variabili è composta da  $N$  osservazioni allora il loro coefficiente di Pearson è definito come:

$$\rho(A, B) = \frac{1}{N-1} \sum_{i=1}^N \left( \frac{A_i - \mu_A}{\sigma_A} \right) \left( \frac{B_i - \mu_B}{\sigma_B} \right) \quad (8.6)$$

dove  $\mu_A$  e  $\sigma_A$  sono rispettivamente la deviazione standard e la media della variabile aleatoria  $A$  e  $\mu_B$  e  $\sigma_B$  sono rispettivamente la deviazione standard e la media della variabile aleatoria  $B$ . Alternativamente è possibile definire il coefficiente di correlazione tra le due variabili come funzione della loro covarianza:

$$\rho(A, B) = \frac{cov(A, B)}{\sigma_A \sigma_B} \quad (8.7)$$

Detto ciò si definisce la matrice dei coefficienti di correlazione per ogni coppia di variabili aleatorie come:

$$R = \begin{pmatrix} \rho_{A,A} & \rho_{A,B} \\ \rho_{B,A} & \rho_{B,B} \end{pmatrix} = \begin{pmatrix} 1 & \rho_{A,B} \\ \rho_{B,A} & 1 \end{pmatrix} \quad (8.8)$$

e dal momento che  $A$  e  $B$  sono sempre direttamente correlate con loro stesse la diagonale sarà identicamente 1. Allo scopo di valutare la bontà della predizione viene fatta la media dei coefficienti di correlazione dei diversi trial dei quali viene calcolata la stima tramite regressione:

$$accuracy_{corrcoef} = \frac{1}{T} \sum_{i=0}^T R_{1,2}(original_i, estimate_i) \in [0, 1] \quad (8.9)$$

dove  $T$  è il numero di trial stimati dal modello, *original* è il segnale di *glove* come appare nel database Ninapro e *estimate* è il relativo segnale stimato.

### 8.3 Analisi dei risultati

Nella fase finale è importante valutare i sottoinsiemi di canali selezionati, restituiti dall'algoritmo. Come già accennato nei capitoli precedenti, la bontà dei canali selezionati dipende indirettamente dall'accuratezza del segnale che il modello genera a fronte dei canali stessi. Concorrono a migliorarne le prestazioni molteplici altri parametri che nel caso particolare del modello GMM utilizzato sono:

- Il numero di canali scelti per allenare il modello
- Il numero di componenti della mistura gaussiana
- Il numero di campioni che compongono i segnali
- I giunti che è chiesto di stimare al modello
- Quanti e quali giunti vengono usati per l'allenamento del modello
- Il tipo di pre-processing ed elaborazioni applicati ai segnali sEMG forniti in ingresso al modello
- Il movimento preso in considerazione

Detto questo, ricopre un ruolo fondamentale anche la misura di accuratezza scelta. In questo elaborato ne sono state utilizzate due: Goodness Of Fit basata su NMSE e coefficienti di correlazione (vedi *sezione 8.2*).

Prendendo a titolo di esempio i risultati ottenuti per il movimento M1 con i seguenti parametri:

- 13 componenti per la mistura gaussiana
- 1405 campioni per ogni segnale

- 6 canali per allenare il modello GMM
- 3 giunti da stimare: #2, #3 e #4
- Trial da 1 a 5 di tutti e 40 i soggetti utilizzati come training set per allenare il modello GMM
- Sesto trial di tutti e 40 i soggetti utilizzati come validation (o test) set per calcolare la regressione

Sono stati ottenute i seguenti risultati:

- Last-6 channels: 4,6,7,9,10,11  $\Rightarrow$  Accuracy = 19.4%
- Top-6 channels: 1,2,3,5,8,12  $\Rightarrow$  Accuracy = 42.6%
- Top-6 channels: 1,2,3,5,8,12 con outliers detection  $\Rightarrow$  Accuracy = 52.6%

Questi risultati dimostrano che l'utilizzo dei 6 peggiori canali della classifica stilata dall'algoritmo di selezione mostra un'accuratezza abbastanza bassa, di meno del 20%. L'utilizzo dei migliori sei canali del ranking porta un miglioramento che fa salire l'indice a oltre i 40 punti percentuale. L'utilizzo degli stessi canali con l'aggiunta del metodo di outlier detection introduce un ulteriore miglioramento facendo oltrepassare il 50% alla soglia di accuratezza. Studi precedenti dimostrano che i risultati ottenuti possono migliorare ulteriormente affinando la fase di pre-processing, ma questo studio esula dallo scopo di questa tesi.

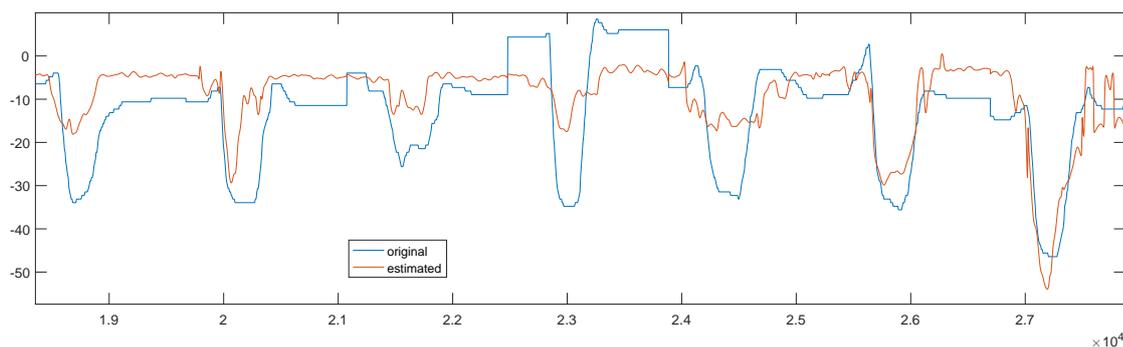


Figura 8.1: Andamento del segnale originale di un giunto e del relativo segnale stimato dal modello GMR per il movimento M1.



# Capitolo 9

## Conclusioni

### 9.1 Discussione

In questa tesi viene proposto un algoritmo per la selezione dei canali più significativi da un sistema di elettromiografia superficiale. L'intero progetto nasce nell'ottica di essere inserito in un contesto preesistente ben descritto da Michieletto et al. in [47], da Valentini et al. in [30] e da Stival et al. in [31]. Questo ha aiutato sin dall'inizio a delineare gli obiettivi di indipendenza dal soggetto e di analisi di segnali multi-giunto, nonché a indirizzare l'intera tesi all'utilizzo di tecniche già adeguatamente documentate per il preprocessing dei segnali sEMG (la Wavelet Transform) come pure per la costruzione di un modello probabilistico sulla base del quale predire movimenti con un algoritmo di regressione a partire da nuovi segnali sEMG (Gaussian Mixture Model e Gaussian Mixture Regression).

L'algoritmo fonda il suo funzionamento sul concetto di somiglianza e riconoscibilità dei segnali, misurata grazie all'uso della covarianza incrociata e della differenza di due segnali. Sulla base di queste due misure sono stati attribuiti dei punteggi opportunamente pesati ad ogni canale i quali, fissato un movimento, concorreranno alla valutazione finale di quel canale. Il ranking dei canali del sistema sEMG per ogni movimento, rappresenta il vero e proprio risultato restituito dall'algoritmo che poi viene passato come ingresso al modello GMM. Avendo a disposizione un modello probabilistico è possibile calcolare il movimento dei giunti a partire dai segnali sEMG dei canali selezionati dall'algoritmo, per mezzo dell'algoritmo GMR. La valutazione dei risultati dell'algoritmo si basano indirettamente sugli indici di accuratezza calcolati confrontando i segnali originali e quelli restituiti dalla regressione.

Tenendo in considerazione le varie prove effettuate e i relativi risultati ottenuti, si può affermare che l'algoritmo descritto costituisce un valido sostituto del metodo "manuale-qualitativo" per la selezione dei canali più promettenti per un approccio probabilistico costruito a partire da segnali biometrici. La possibilità di poter contare su un metodo automatizzato che assista l'utente nella scelta dei canali sEMG più significativi risulta estremamente utile nel caso in cui si volesse aumentare il numero dei movimenti da riconoscere e il numero dei soggetti che compongono il dataset.

## 9.2 Sviluppi futuri

L'algoritmo è suddivisibile in 4 fasi principali: estrazione dei dati e pre-processing, outlier detection, elaborazione dei risultati ed infine calcolo dell'accuratezza delle predizioni. Ognuna di queste fasi è stata studiata ed analizzata in maniera indipendente, pertanto, può subire modifiche senza incidere sul corretto funzionamento delle altre. Ogni parte espone un problema risolvibile in molteplici modi e questo lascia spazio ad ulteriori ricerche ed approfondimenti che potrebbero portare ad un miglioramento dello stato dell'arte e del modello descritto fin qui. Una possibilità potrebbe essere quella di estendere le sperimentazioni ad ambiti differenti seppur legati agli stessi obiettivi.

Considerando la fase di estrazione dei dati e l'elevato numero di movimenti presenti nel dataset Ninapro, una semplice espansione del lavoro attuale è quella di applicare l'algoritmo ai segnali sEMG relativi ad altri movimenti o a soggetti non trattati da questa tesi.

Nella fase di pre-processing dei segnali ed in particolare in quello di outlier detection, le nuove strade percorribili sono numerose. Avendo a disposizione 240 segnali ( $6 \text{ trial} \times 40 \text{ soggetti}$ ) fissati un movimento ed un canale, è possibile concatenare tutti questi segnali ottenendo una time-series. Questo tipo di oggetti vanta numerose ricerche a riguardo, che trattano metodi per la loro elaborazione, analisi e ricerca delle anomalie. Per citarne alcuni: AR, MA, ARMA, ARIMA, ARIMAX sono tutti modelli autoregressivi e a media mobile, spesso usati come base sulla quale applicare tecniche di outlier detection.

Nella fase di elaborazione dell'algoritmo sono investigabili altrettante modifiche, tra le quali potrebbero essere considerate il calcolo della somiglianza tra segnali basato su PCA in più di 2 dimensioni alla quale applicare indici di aggregazione tipici della Cluster Analysis. Avendo a disposizione diverse misure di somiglianza tra segnali si può pensare di unirle in un unico punteggio di somiglianza dando loro pesi diversi e sulla base di questi valutare i canali più significativi.

Per quanto riguarda la quarta fase, quella di calcolo dell'accuratezza, una possibilità di miglioramento consiste nell'estendere l'analisi già fatta al più alto numero di combinazioni di giunti possibile, per poterne confrontare l'accuratezza. Si potrebbe inoltre considerare la creazione di un algoritmo che svolga un compito analogo a quello descritto in questa tesi ed esegua la selezione dei giunti più significativi per un determinato movimento. Guardando agli stessi segnali nel campo della statistica multivariata, è possibile valutarli su un numero di dimensioni maggiore delle 2 considerate in alcuni degli outlier detector descritti in questa tesi. In questo contesto andrebbero valutati in diversi metodi di anomaly detection o più in generale di unsupervised learning. Alcuni di quelli testati ma non utilizzati sono la tecnica di Kernel Density Estimation (KDE) o Cluster Analysis su dati multidimensionali.



# Bibliografia

- [1] S. J. Orfanidis. *Optimum Signal Processing: An Introduction. 2nd Edition.* McGraw-Hill, 1996.
- [2] J. Larsen. *Correlation Functions and Power Spectra.* 2009.
- [3] H. Liu, S. Shah, and W. Jiang. *On-line outlier detection and data cleaning.* Computers and Chemical Engineering, Vol. 28, 2004.
- [4] D.J. Olive. *A resistant estimator of multivariate location and dispersion.* Computational Statistics and Data Analysis, Vol. 46, 2004.
- [5] D. J. Olive, and D. M. Hawkins. *Robustifying Robust Estimators.* 2007.
- [6] P. J. Rousseuw, and K. Van Driessen. *A fast algorithm for the minimum covariance determinant estimator.* Technometrics, Vol. 41, 1999.
- [7] G. Pison, S. Van Aelst, and G. Willems. *Small Sample Corrections for LTS and MCD.* Metrika, Vol. 55, 2002.
- [8] R. Maronna, and R. H. Zamar. *Robust estimates of location and dispersion for high dimensional datasets.* Technometrics, Vol. 50, 2002.
- [9] R. K. Pearson, Y. Neuvo, J. Astola, and M. Gabbouj. *Generalized Hampel Filters.* EURASIP Journal on Advances in Signal Processing, 2016.
- [10] L. Davies, and U. Gather. *The identification of multiple outliers.* Journal of the American Statistical Association, Vol. 88, 1993.
- [11] M. Krzywinski, and N. Altman. *Points of Significance: Visualizing samples with box plots.* Nature Methods, Vol. 11, 2014.
- [12] P. J. Huber, and E. M. Ronchetti. *Robust Statistics, Second Edition.* John Wiley & Sons, Inc., 2009.

- [13] M. Hubert, and M. Debruyne. *Minimum covariance determinant*. WIREs Computational Statistics, John Wiley & Sons, Inc., 2010.
- [14] C. Fauconnier, and G. Haesbroeck. *Outliers detection with the minimum covariance determinant estimator in practice*. Statistical Methodology, Vol. 6, 2009.
- [15] C. Leys, C. Ley, O. Klein, P. Bernard, and L. Licata. *Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median*. Journal of Experimental Social Psychology, 2013.
- [16] R. J. Hyndman and H. L. Shang. *Rainbow plots, bagplots and boxplots for functional data*. Journal of Computational and Graphical Statistics, Vol. 19, pp. 29-45, 2010.
- [17] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. the MIT Press, 2006.
- [18] P. Filzmoser, S. Aivazian, P. Filzmoser, and Y. Kharin. *A multivariate outlier detection method*. Proceedings of the Seventh International Conference on Computer Data Analysis and Modeling, Vol. 1, pp. 18-22, 2004.
- [19] M. Steinbach, V. Kumar, P.N. Tan. *Introduction to Data Mining*. Pearson, 2005.
- [20] V. Chandola, A. Banerjee, and V. Kumar. *Anomaly detection: A survey*. ACM Computing Surveys, 2009.
- [21] M. Goldstein, and S. Uchida. *A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data*. PLoS ONE, 2016.
- [22] V.J. Hodge, and J. Austin. *A survey of outlier detection methodologies*. Artificial Intelligence Review, pp. 85-126, 2004.
- [23] V. Chandola, A. Banerjee, and V. Kumar. *Anomaly Detection for Discrete Sequences: A Survey*. IEEE Transactions on Knowledge and Data Engineering, Vol. 24, 2012.
- [24] T. Shibanoki, K. Shima, T. Takaki, Y. Kurita, A. Otsuka, T. Chin, and T. Tsuji. *Multi-channel Surface EMG Classification Based on a Quasi-optimal Selection of Motions and Channels*. ICME International Conference on Complex Medical Engineering, 2012.

- [25] Y. Geng, X. Zhang, Y. Zhang, and G. Li. *A novel channel selection method for multiple motion classification using high-density electromyography*. BioMedical Engineering OnLine, 2014.
- [26] A. Al-Ani, I. Koprinska, G. R. Naik, and R. N. Khushaba. *A Dynamic Channel Selection Algorithm for the Classification of EEG and EMG Data*. International Joint Conference on Neural Networks, 2016.
- [27] J. Liu, X. Li, G. Li, and P. Zhou. *EMG Feature Assessment for Myoelectric Pattern Recognition and Channel Selection: A Study with Incomplete Spinal Cord Injury*. Med Eng Phys, 2014.
- [28] R. H. Chowdhury, M. B. I. Reaz, M. A. B. M. Ali, A. A. A. Bakar, K. Chellappan, and T. G. Chang. *Surface Electromyography Signal Processing and Classification Techniques*. Sensors, vol. 13, 2013.
- [29] D.K. Kumar, N.D. Pah, and A. Bradley. *Wavelet analysis of surface electromyography*. IEEE Transactions on Neural Systems and Rehabilitation Engineering, Vol. 11, 2003.
- [30] R. Valentini, S. Michieletto, F. Spolaor, Z. Sawacha, and E. Pagello. *Processing of sEMG Signals for Online Motion of a Single Robot Joint through GMM Modelization*. IEEE International Conference on Rehabilitation Robotics (ICORR), 2015.
- [31] F. Stival, S. Michieletto, and E. Pagello. *Online subject-independent modeling of sEMG signals for the motion of a single robot joint*. 6<sup>th</sup> IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob), 2016.
- [32] L. Chun-Lin. *A Tutorial of the Wavelet Transform*. 2010.
- [33] M. Atzori, A. Gijsberts, S. Heynen, A. M. Hager, O. Deriaz, P. van der Smagt, C. Castellini, B. Caputo, and H. Müller. *Building the NINAPRO Database: A Resource for the Biorobotics Community*. The Fourth IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics, 2012.
- [34] M. Atzori, A. Gijsberts, I. Kuzborskij, S. Elsig, A. M. Hager, O. Deriaz, C. Castellini, H. Müller, and B. Caputo. *Characterization of a Benchmark Database for Myoelectric Movement Classification*. IEEE Transactions on Neural Systems and Rehabilitation Engineering, Vol. 23, 2015.

- [35] A. Gijsberts, M. Atzori, C. Castellini, H. Müller, and B. Caputo. *The Movement Error Rate for Evaluation of Machine Learning Methods for sEMG-based Hand Movement Classification*. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 2014.
- [36] M. Atzori, A. Gijsberts, C. Castellini, B. Caputo, A. M. Hager, S. Elsig, G. Giatsidis, F. Bassetto, and H. Müller. *Electromyography data for non-invasive naturally-controlled robotic hand prostheses*. Scientific Data 1, 2014.
- [37] S. Lhermitte, J. Verbesselt, W.W. Verstraeten, and P. Coppin. *A comparison of time series similarity measures for classification and change detection of ecosystem dynamics*. Remote Sensing of Environment, 2011.
- [38] J. Serrà, and J. L. Arcos. *An empirical evaluation of similarity measures for time series classification*. Artificial Intelligence Research Institute (IIIA-CSIC), Spanish National Research Council, 2014.
- [39] F. Iglesias, and W. Kastner. *Analysis of Similarity Measures in Times Series Clustering for the Discovery of Building Energy Patterns*. Energies, 2013.
- [40] A. P. Dempster, N. M. Laird, and D. B. Rubin. *Maximum-Likelihood from incomplete data via the EM algorithm*. Journal of the Royal Statistical Society, 1977.
- [41] Y. Huang, K. B. Englehart, B. Hudgins, and A. D. Chan. *A gaussian mixture model based classification scheme for myoelectric control of powered upper limb prostheses*. IEEE Transactions on Biomedical Engineering, vol. 52, 2005.
- [42] A. D. Chan and K. B. Englehart. *Continuous myoelectric control for powered prostheses using Hidden Markov Models*. IEEE Transactions on Biomedical Engineering, vol. 52, 2005.
- [43] E. J. Scheme, K. B. Englehart, and B. S. Hudgins. *Selective classification for improved robustness of myoelectric control under nonideal conditions*. IEEE Transactions on Biomedical Engineering, vol. 58, 2011.
- [44] J. Han, Q. Ding, A. Xiong, and X. Zhao. *A state space emg model for the estimation of continuous joint movements*. 2014.
- [45] K. Manal, R. V. Gonzalez, D. G. Lloyd, and T. S. Buchanan. *A realtime EMG-driven virtual arm*. Computers in Biology and Medicine, vol. 32, 2002.

- [46] Y. Chen, X. Zhao, and J. Han. *Hierarchical projection regression for online estimation of elbow joint angle using EMG signals*. Neural Comput and Applic, vol. 23, 2013.
- [47] S. Michieletto, M. Antonello, L. Tonin, R. Bortoletto, F. Spolaor, E. Pagello, and E. Menegatti. *GMM-based single-joint angle estimation using EMG signals*. Proceedings of the 13th Conference on Intelligent Autonomous Systems (IAS-13), 2014.
- [48] K. Pearson. *On Lines and Planes of Closest Fit to Systems of Points in Space*. Philosophical Magazine, Vol. 2, 1901.
- [49] H. Hotelling. *Analysis of a complex of statistical variables into principal components*. Journal of Educational Psychology, Vol. 24, 1933.
- [50] H. Hotelling. *Relations between two sets of variates*. Biometrika, Vol. 28, 1936.
- [51] J. Li, J. A. Cuesta-Albertos, and R. Y. Liu. *DD-Classifier: Nonparametric Classification Procedure Based on DD-plot*. 2013.



# Ringraziamenti

Anche questo percorso è giunto al termine. Gli anni di università mi hanno insegnato molto dal punto di vista accademico, portandomi a conoscere discipline affascinanti che spero di approfondire nel mondo del lavoro. Ritengo di essere fortunato per aver avuto il privilegio di studiare in un ambiente così stimolante e a stretto contatto con persone fantastiche.

Per questo ringrazio la mia famiglia, in particolare mia mamma per avermi appoggiato in ogni mia scelta e mio papà che, a modo suo, mi ha sempre aiutato per tutta la durata di questo lungo cammino.

Ringrazio Natalia per essermi stata accanto dall'inizio di questa avventura, per aver festeggiato con me i traguardi raggiunti e per avermi sostenuto durante i momenti più difficili.

Ringrazio gli amici, in particolare quelli del gruppo di università, che hanno condiviso con me le gioie e le fatiche della vita accademica.

Un ultimo ringraziamento a tutti coloro che in questi anni, con piccoli e grandi contributi, mi hanno aiutato a raggiungere questa importante meta.