



UNIVERSITÀ  
POLITECNICA  
DELLE MARCHE

FACOLTÀ DI INGEGNERIA

CORSO DI LAUREA IN INGEGNERIA INFORMATICA E DELL'AUTOMAZIONE

---

# **Paradigma delle BCI applicato ad un contesto di realtà virtuale**

## **BCI paradigm applied to a virtual reality context**

Candidato:

**Edoardo Bilancia**

Relatore:

**Prof. Simone Fiori**

Anno Accademico 2020-2021

# Ringraziamenti

Una menzione particolare al mio relatore, Prof. Simone Fiori, il quale malgrado l'attuale condizione pandemica ha sempre fornito la sua disponibilità a ogni mia richiesta prima e durante la stesura della tesi, e nella fiducia riposta nelle mie abilità.

Ringrazio inoltre il prof. Pasquale Scilingo e il Prof. Giovanni Vozi per l'esclusiva opportunità offerta di frequentare il Centro di ricerca E. Piaggio, luogo in cui ho potuto affrontare e sviluppare i contenuti di questa tesi.

Infine, si ringrazia per il prezioso supporto, il Dott. Alejandro Luis Callara, il prof. Alberto Greco, e tutti coloro i quali sono stati d'aiuto nel corso dell'Internship.

*Ancona, Giugno 2021*

Edoardo Bilancia



# Sommario

Le BCI offrono l'opportunità, in pazienti con patologie, di recupero delle capacità motorie attraverso la decodifica e la conversione dell'attività cerebrale in segnali di controllo. Parallelamente alle tecniche di neuroimaging, che permettono l'acquisizione dell'attività cerebrale, Gli algoritmi di machine learning offrono sistemi di estrapolazione dell'informazione neurale.

Il focus di questa tesi è incentrato nella descrizione e nello sviluppo di un'interfaccia neurale asincrona capace di discriminare due classi di eventi, con presentazione del feedback visivo attraverso l'ambiente di realtà virtuale.

Dopo una introduzione al concetto delle BCI nei primi capitoli, verranno affrontate, passo dopo passo, le procedure svolte per lo sviluppo dell'interfaccia neurale, con approfondita trattazione analitica dei principali algoritmi integrati e del codice utilizzato per lo sviluppo dell'ambiente virtuale.



# Indice

<b>1</b>	<b>Le BCI</b>	<b>1</b>
1.1	Introduzione . . . . .	1
1.2	Obiettivi . . . . .	1
1.3	Discriminazione degli eventi . . . . .	2
1.4	EEG-Bci . . . . .	3
1.4.1	Il sistema internazionale 10-20 . . . . .	3
1.4.2	Ulteriori tecniche di Neuroimaging . . . . .	4
1.4.3	tecniche di Neuroimaging - approcci ibridi . . . . .	5
1.5	ERD/ERS . . . . .	6
1.6	Paradigmi ricorrenti nelle BCI . . . . .	6
<b>2</b>	<b>Neurofisiologia e ritmi cerebrali</b>	<b>9</b>
2.1	La corteccia motoria . . . . .	9
2.1.1	La corteccia motoria primaria - Area M1 . . . . .	10
2.1.2	La corteccia premotoria . . . . .	10
2.1.3	La corteccia motoria supplementare . . . . .	11
2.2	Ritmi cerebrali . . . . .	11
<b>3</b>	<b>Software a supporto</b>	<b>13</b>
3.1	Acquisizione del dato . . . . .	13
3.1.1	Trasmissione del segnale - DSI-Streamer . . . . .	13
3.1.2	Tasmissione del segnale - Lab Streaming Layer . . . . .	14
3.2	Processamento e analisi del dato - EEGLab . . . . .	15
3.3	Ambiente di realtà virtuale - Unity3D . . . . .	15
<b>4</b>	<b>setup di acquisizione</b>	<b>17</b>
4.1	Hardware per l'acquisizione . . . . .	17
4.2	Video di training . . . . .	18
<b>5</b>	<b>Preparazione e modifica del dato</b>	<b>19</b>
5.1	Caricamento e gestione delle acquisizioni . . . . .	19
5.2	Filtraggio del dato . . . . .	20
5.3	Pulizia del tracciato EEG . . . . .	20
5.3.1	Rimozione dei segmenti compromessi . . . . .	21

<b>6</b>	<b>BCILAB, pre-processing e classificazione</b>	<b>23</b>
6.1	CSP - Common Spatial Pattern . . . . .	23
6.1.1	Procedure matematiche . . . . .	24
6.2	Un introduzione ai classificatori . . . . .	25
6.2.1	Iper-parametri . . . . .	26
6.2.2	Prestazione dei classificatori . . . . .	26
6.2.3	Sensibilità e Specificità . . . . .	26
6.2.4	Le curve ROC . . . . .	27
6.3	LDA - Linear Discriminant Analysis . . . . .	28
6.4	BCILAB . . . . .	32
6.4.1	Analisi Offline . . . . .	32
6.4.2	Analisi Online . . . . .	32
<b>7</b>	<b>Ambiente di realtà virtuale</b>	<b>35</b>
7.1	Unity3D . . . . .	35
7.1.1	Socket TCP/IP . . . . .	36
7.1.2	Parsing del dato . . . . .	37
7.1.3	Animazioni e Animator Controller . . . . .	38
<b>8</b>	<b>Conclusioni e prospettive future</b>	<b>39</b>

# Capitolo 1

## Le BCI

### 1.1 Introduzione

le BCI (Brain-Computer-interface) sono ormai realtà, le potenzialità legate a questa emergente tecnologia e le prospettive con le quali si pone la rendono una ambiziosa tematica di studio. I recenti avanzamenti nel campo biomedico, informatico e neuroscientifico hanno permesso l'affermazione di questa tecnologia in un ottica che si propone d'incrementare notevolmente le capacità fisiche e mentali delle persone, patologiche o meno, strutturando un canale di comunicazione tra il cervello e le macchine [1]. Le possibilità messe a disposizione da questa tecnologia sono diverse, i campi più studiati spaziano da applicazioni per l'assistenza di pazienti patologici a sistemi di supporto per compiti che risultano complessi e monotoni. Diverse aziende si stanno orientando verso lo sviluppo di questa innovativa tecnologia rimarcando le innumerevoli possibilità realizzabili, particolarmente in ambito medico, dove si prospetta di trovare rimedio a patologie rare che comportano nei pazienti un'impossibilità d'interfacciamento con altri, e con l'ambiente circostante [2].

### 1.2 Obiettivi

Fin dalla nascita delle prime GUI (Graphical User Interface) si è cercato di offrire il miglior metodo d'interazione che permettesse alle persone il corretto utilizzo di un sistema nascondendone le complessità, attraverso le Brain Computer Interfaces siamo in grado di raggiungere un nuovo macro stadio d'interazione, che non coinvolge più l'utilizzo del sistema motorio per interagire con l'ambiente circostante. Si cerca invece di coinvolgere l'attività neurale che garantisce e si pone nell'ipotesi, in alcune applicazioni, di poter fornire una maggiore velocità di risposta e un controllo incondizionato. Un esempio può essere quello del segnale ERRp, ovvero il potenziale evocato nel momento in cui si presenta un "errore" nel compimento di un task [3]. Il focus di questa tesi è incentrato nello studio e nello sviluppo di un sistema BCI, completato da un applicativo software in grado di fornire all'utente un'esperienza di realtà virtuale, in cui il soggetto è in grado di eseguire delle azioni attraverso il solo utilizzo dell'immaginazione.

### 1.3 Discriminazione degli eventi

I processi mentali che coinvolgono l'attivazione di atti motori scaturiscono nel cervello, o più in particolare lungo la corteccia motoria, delle variazioni in termini di potenza spettrale che possono essere studiati e analizzati al fine di acquisire l'effettiva volontà del paziente.

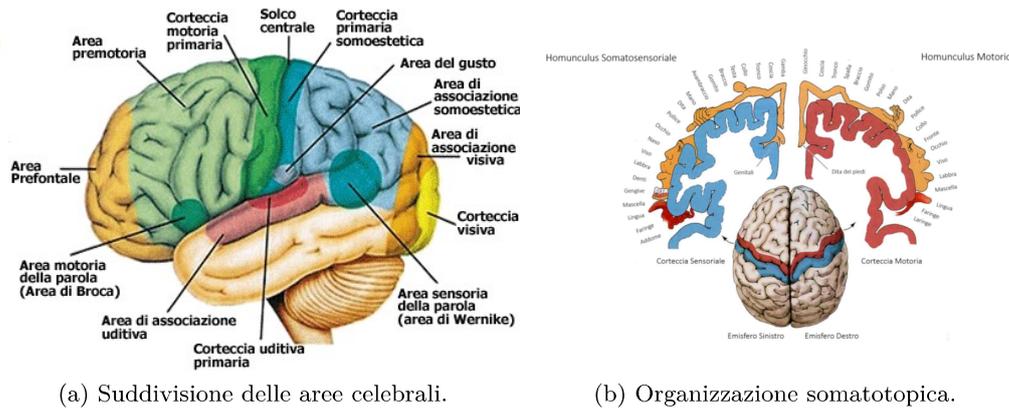


Figura 1.1: Organizzazione del cervello

Benché la corteccia motoria è responsabile del controllo di tutto il sistema motorio, la rappresentazione dei vari arti all'interno del solco non è omogeneo, come rappresentato in figura 1.1b, per questo si ricorre spesso in questi studi di MI-EEG based, al movimento di arti che coinvolgono l'attivazione di un gruppo esteso di neuroni in quanto poi la risoluzione spaziale dell'EEG viene inficiata dalla conduzione di volume nel mezzo.

L'applicazione di filtri spaziali, ad esempio il Common Spatial Pattern (CSP) come tecnica di filtraggio del segnale, che sarà tratta nel capitolo 6, comporta la costruzione di una matrice di mix  $W$  tale da applicare una trasformazione lineare che proietta il segnale originale  $X$  ( $N$ canali  $\times$  timesamples) in un sottospazio che si distingue dal primo per il numero minore di sorgenti, in particolare la matrice  $W$  è costituita da valori di peso per ogni canale, in questo modo le sorgenti risultanti  $S$  saranno un mix dei segnali di partenza pesati in base alla variazione di potenza nelle bande di frequenza d'interesse [4]

$$S = WX \tag{1.1}$$

In seguito a questa trasformazione, vengono estratte le features che discriminano le due classi, nel caso della Motor Imagery la grandezza che viene misurata è la potenza spettrale, che sappiamo sarà caratterizzata da una soppressione in ampiezza nel momento in cui si verifica l'evento, tale occorrenza prende il nome di Event Related Desynchronization (ERD). L'incremento in ampiezza nelle stesse bande di frequenza nel momento in cui l'evento termina, prende invece il nome di Event related Synchronization (ERS) [5].

Il classificatore che poi si occuperà di costruire il piano di divisione tra le due classi di eventi sarà l'LDA (linear discriminant analysis) che sarà affrontato nel capitolo 6.

## 1.4 EEG-Bci

L'acquisizione del dato di partenza con il quale vorremo conformare la nostra particolare applicazione rappresenta lo stadio di partenza per ogni studio di questo tipo, esistono diverse tecniche di acquisizione che si distinguono per risoluzione spaziale e temporale. L'EEG è la tecnica di acquisizione più semplice e ricorrente in quanto gli elettrodi sono applicati direttamente sullo scalpo del partecipante. Attraverso questa applicazione, vengono estratti i potenziali elettrici cerebrali generati dalle cellule neuronali sulle quali, scorrendo una corrente extracellulare, si viene a generare una differenza di potenziale misurabile sulla superficie dello scalpo. Benché l'EEG sia una delle tecniche più utilizzate, quest'ultima soffre di problemi di SNR per via della conduzione di volume dovuta alla conformazione del cranio e del cervello. Per questo motivo, l'EEG offre un'ottima risoluzione temporale, in compenso la risoluzione spaziale è minore rispetto ad altri sistemi di acquisizione come ad esempio Fmri.

### 1.4.1 Il sistema internazionale 10-20

Il sistema internazionale 10-20 rappresenta uno degli standard di posizionamento degli elettrodi più ricorrenti. Introdotta da Jasper [6], questa tecnica prevede l'applicazione di 21 elettrodi sullo scalpo del paziente in posizioni ben definite, lungo cinque linee trasversali e distanziate tra loro del 10% o 20%, visibili in Figura 1.2

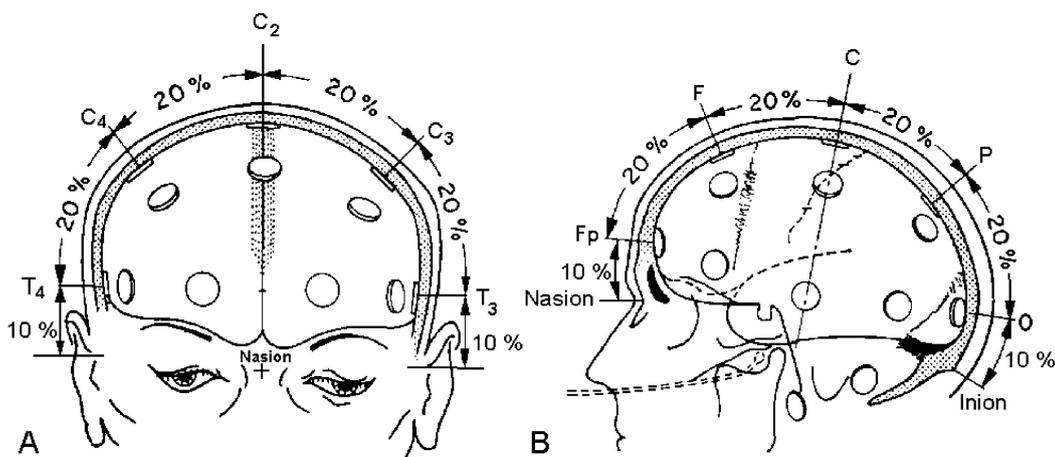


Figura 1.2: Immagine originale dello standard proposto da Jasper, erroneamente gli elettrodi sono posizionati direttamente sulla corteccia e non sullo scalpo

La nomenclatura degli elettrodi è definita da lettere e numeri che ne identificano per ognuno la posizione. Gli elettrodi aventi numeri pari sono posizionati sopra l'emisfero destro, viceversa, gli elettrodi aventi numero dispari sono posizionati nell'emisfero opposto. partendo dalla fronte del soggetto le lettere si susseguono in questo ordine: "Fp" Pre-frontale, "F" frontale, "C" centrale, "P" parietale, "O" occipitale e infine "T" temporale.

Per una spiegazione più esaustiva relativa al sistema 10-20 ed ai sistemi derivanti da questo fare riferimento a [7]

### 1.4.2 Ulteriori tecniche di Neuroimaging

Come introdotto nel paragrafo precedente, l'acquisizione del segnale fisiologico rappresenta una delle fasi fondamentali di sviluppo, caratterizzato dalla sua semplicità di utilizzo l'EEG domina la scena nel campo delle BCI, eppure ulteriori tecniche di neuroimaging si stanno affermando in questo campo nell'ottica di migliorare la qualità dell'acquisizione. Di seguito alcuni sistemi di acquisizione dei segnali cerebrali:

- ECoG (Electrocorticography) o IEEG (inter-cranial electroencephalography): Questa tecnica prevede che gli elettrodi vengano applicati al di sotto del cranio, a contatto con quella che è la corteccia cerebrale. Benché sia la risoluzione spaziale e temporale è migliore rispetto alla più classica applicazione sullo scalpo, il problema risulta proprio il tipo di ambiente necessario per applicare questi elettrodi direttamente sulla corteccia del cervello, rendendola una tecnica altamente invasiva e quindi poco applicabile in molti ambiti [8].
- MEG (magnetoencephalography): tecnica che prevede la misurazione del campo magnetico generato dalle correnti interne all'encefalo. Impiega dei sensori denominati SQUID (Superconducting quantum interference device) in grado di acquisire variazioni nel campo magnetico nell'ordine dei femtoTesla ( $10^{-15}$  T). Al fine di raggiungere un tale valore di precisione è necessario applicare questa tecnica in un ambiente controllato e altamente specializzato, il che la rendono una tecnica poco ricorrente nel campo delle BCI.

A differenza delle precedenti tecniche discusse, le quali si basano sulle radiazioni elettromagnetiche per l'analisi dell'attività cerebrale, le tecniche di seguito riportate misurano le variazioni del flusso ematico conseguente all'aumento di attività neuronale.

- Fmri (functional magnetic resonance imaging): È una Tecnica di neuroimaging che si basa sulle caratteristiche magnetiche dell'emoglobina, in particolare si osserva come quest'ultima varia in funzione del livello di ossigenazione del sangue in particolari zone del cervello. Attraverso l'utilizzo del segnale

BOLD (Blood Oxygenation Level Dependent) si osservano le variazioni del flusso sanguigno cerebrale, in rispetto al consumo d'ossigeno. Un segnale di BOLD maggiore si verifica nella condizione di maggiore afflusso sanguigno in proporzione al consumo di ossigeno, viceversa, un segnale di BOLD minore è caratterizzato da un afflusso di sangue minore in proporzione alle variazioni di consumo di ossigeno [9]. La bassa risoluzione temporale non ne garantiscono un'applicazione attiva nel campo delle BCI, in compenso la tecnica discussa viene utilizzata per l'analisi fisiologica dei segnali cerebrali.

- PET (Positron Emission Tomography): La PET rappresenta un'altra tecnica di acquisizione di segnali fisiologici cerebrali, la sua applicazione nel campo attivo delle BCI è subito inficiata dalla necessità di dover somministrare al paziente radioisotopi che fungono da tracciatori per lo scanner tomografo, l'utilizzo più ricorrente ricade nell'ambito oncologico.
- NIRS (Near InfraRed Spectroscopy) Utilizza la radiazione ottica nella banda spettrale del vicino infrarosso per "colpire" il paziente nell'area d'interesse da un fotone NIR, che, attraversando il tessuto, si diffonde seguendo un percorso non rettilineo. L'analisi del percorso seguito attraverso parametri di scattering e assorbimento permette lo studio delle variazioni del livello di ossigenazione cerebrale. Recentemente sempre più utilizzata nel campo delle BCI per via della facilità d'applicazione sul soggetto e per la possibilità di utilizzo di questa tecnica in contemporanea con altri sistemi di acquisizione [10].

### 1.4.3 tecniche di Neuroimaging - approcci ibridi

Come già menzionato, le tecniche di acquisizione dei segnali cerebrali possono essere caratterizzate in termini di risoluzione spaziale e temporale, ogni tipo di approccio prevede l'utilizzo di una particolare tecnica di acquisizione in base al tipo di risultato che si vuole ottenere. L'utilizzo di più tecniche di acquisizione che mirano a ottenere lo stesso risultato portano a vantaggi in termini di accuratezza di classificazione, oltre a permettere un maggior numero di azioni di controllo. Approcci ibridi come EEG + NIRS [10] oppure EEG + EOG [11] sono i più ricorrenti nel campo delle BCI. Come descritto, tutte queste metodologie di acquisizione hanno pro e contro, esiste però un notevole margine di miglioramento riguardo l'acquisizione attraverso EEG. Tecniche come ICA (Independent Component Analysis) [12] o PCA (Principal Component Analysis), applicano delle trasformazioni nello spazio dei dati per massimizzare l'informazione estratta, mentre tecniche di Beamforming [13] permettono l'utilizzo del sistema di acquisizione in maniera più accurata, estrapolando informazioni da precise regioni del cervello. Tali tecniche applicano quindi un post-processing del dato che diviene quindi molto più accurato in termini spaziali andando a ovviare al problema della dispersione del segnale nel mezzo conduttore, aumentando la risoluzione spaziale e l'SNR.

## 1.5 ERD/ERS

Il focus di questa tesi è incentrato sullo sviluppo di un sistema BCI asincrono che sia in grado di discriminare l'immaginazione della mano sinistra o destra da parte dell'utente. Affinché questo risultato sia raggiungibile, dobbiamo innanzitutto cercare di comprendere come questo fenomeno si traduce all'interno del cervello, o meglio, che interpretazione diamo all'occorrere dell'immaginazione del movimento di un arto. Entrando nel dettaglio della tecnica di Motor Imagery [14] è noto che quest'ultima è legata a un evento di desincronizzazione (ERD) che si manifesta nella zona cerebrale dell'arto immaginato, lungo la sezione che parte da un orecchio e termina sull'opposto, come visto in figura Figura 1.1b. La desincronizzazione comporta una soppressione delle onde MU nella sezione di spazio che interessa l'arto immaginato. Gli eventi ERD/ERS sono definiti come "not phased-locked" in quanto fortemente dipendenti dal contenuto frequenziale. È possibile analizzare, in termini di potenza, lo spettro delle frequenze delle onde  $\alpha$  (7 - 12 Hz) e  $\beta$  (16 - 30 Hz) osservando che, al manifestarsi dell'evento, si presenta una attenuazione in termini di potenza nelle bande interessate. L'ERD/ERS è definita rispettivamente come la percentuale di decremento o incremento della potenza, in accordo con quanto riportato in Eq. 1.2

$$ERD = (A - R)/R * 100 \quad (1.2)$$

Dove  $\mathbf{A}$  è la potenza registrata nell'attività, e  $\mathbf{R}$  è la potenza di riferimento registrata prima dell'evento. Per un'analisi approfondita su ERD/ERS per task di Motor Imagery fare riferimento a [5].

## 1.6 Paradigmi ricorrenti nelle BCI

Le modalità d'interazione con l'utente che permettono la produzione di un segnale di controllo sono diverse, oltre ai già discussi ERD/ERS, legati alla MI (Motor Imagery), esistono diversi segnali elettro corticali che possono essere distinti nel corso dell'acquisizione. Gli ERP (Event-Related Potentials) sono una famiglia di potenziali a basso voltaggio che si verificano in risposta a stimoli o eventi presentati al paziente [15]. Tra i più studiati si presentano la P300 e SSVep (Steady-state Visual Evoked Potential), visibili in figura 1.3.

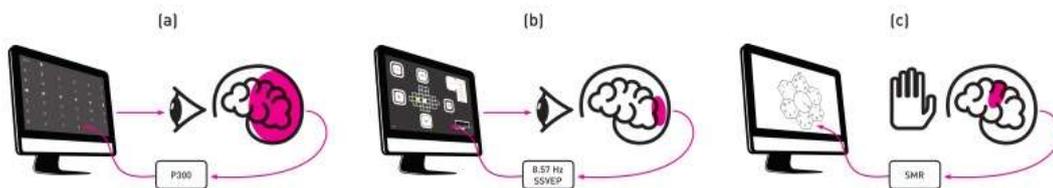


Figura 1.3: L'immagine riporta lo sviluppo di speller, una delle applicazioni maggiormente studiate nel campo delle BCI, attraverso i tre diversi paradigmi più ricorrenti. (a)p300: si sfrutta "oddball paradigm" il che consiste nel presentare uno stimolo raro e distinto rispetto ad altri, la visualizzazione da parte del paziente di questo evento raro genera una deflessione positiva circa 300ms dopo lo stimolo. (b)SSVEP: (Steady-State Visual Evoked Potential) vengono mostrate al paziente diverse possibili selezioni, ognuna di queste caratterizzata da una frequenza di lampeggiamento diversa dalle altre. Il paziente concentrandosi su una di queste frequenze genera un SSVEP alla stessa frequenza nella corteccia visiva (c)Motor Imagery, l'immaginazione da parte dell'utente del movimento di un arto genera un cambiamento nello spettro del segnale nella banda di frequenza 8-30Hz. fonte: [16]



## Capitolo 2

# Neurofisiologia e ritmi cerebrali

### 2.1 La corteccia motoria

La corteccia motoria rappresenta il punto di partenza del nostro studio sui paradigmi delle BCI, in particolare, essendo il focus di questa tesi basato sulla tecnica di Motor Imagery, scendiamo nel dettaglio riguardo la Corteccia motoria, area del cervello adibita al controllo dei movimenti. La corteccia motoria è principalmente composta da 3 sottoparti che possono essere identificate nel lobo frontale, immediatamente anteriori al solco centrale e alla corteccia sensoriale.

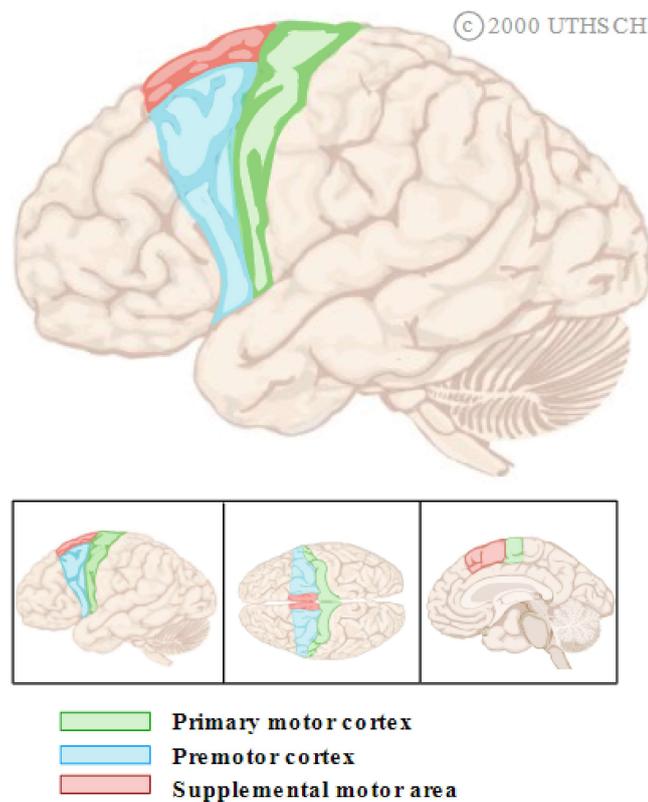


Figura 2.1: fonte: University of Texas

In Figura 2.1 sono rappresentate la corteccia motoria primaria, la corteccia premotoria e la corteccia motoria supplementare. Sebbene la corteccia motoria si occupa di controllare i movimenti, quest'ultima non è adibita alla decisione del movimento da compiere, decisione che viene presa da altre zone del cervello, come la corteccia parietale posteriore e alcune aree della corteccia prefrontale. Secondo la rappresentazione citoarchitettonica di K.Broadmann [17], la corteccia motoria primaria rappresenta l'area 4, mentre la corteccia premotoria e la corteccia motoria supplementare sono entrambe identificate come area 6. Tuttavia, tale suddivisione non può essere rigidamente definita in quanto ogni area cerebrale si integra e si coordina con le altre. Purché la conoscenza di queste aree e la loro attitudine non sia ancora del tutto nota, una breve introduzione al concetto può aiutare a comprendere come catturiamo dal cervello l'input in un sistema BCI.

### **2.1.1 La corteccia motoria primaria - Area M1**

In generale, la corteccia motoria primaria, non controlla direttamente i movimenti di ogni muscolo singolarmente, al contrario quest'ultima sembra codificare singoli movimenti o sequenze di movimenti che coinvolgono più gruppi muscolari. I neuroni della corteccia motoria primaria in particolare codificano diverse informazioni relative al movimento da compiere, difatti quest'ultimi per 'inviare' un movimento agli strati più bassi della gerarchia motoria devono fornire informazioni relative alla velocità piuttosto che le forza da applicare. In particolare, i neuroni della corteccia motoria primaria codificano diverse informazioni utili all'atto del movimento:

- I neuroni della corteccia motoria primaria inoltrano i segnali di movimento ai neuroni Alpha (adibiti alla contrazione del muscolo) 5-100ms prima del movimento.
- La corteccia motoria primaria codifica la forza con la quale deve essere contratto il muscolo.
- Alcuni dei neuroni della corteccia motoria primaria sono adibiti alla codifica della direzione del movimento.
- Infine, alcuni neuroni della corteccia motoria primaria codificano la velocità e l'estensione del muscolo.

### **2.1.2 La corteccia premotoria**

A differenza della corteccia motoria primaria, la corteccia premotoria sembra codificare movimenti più complessi ed è adibita soprattutto alla pianificazione di questi. Una scoperta rilevante nel campo delle BCI basate su Motor Imagery nasce

da un gruppo di neuroscienziati coordinati dal prof. G. Rizzolatti dell'università di Parma, i quali pubblicano uno studio sugli eventi motori [18] in cui viene osservato come le aree cerebrali della corteccia premotoria frontale delle scimmie, responsabili dell'attività motoria, vengano eccitate nella condizione in cui non siano le stesse ad attuare il movimento ma anche nel caso in cui quest'ultime lo osservino. I neuroni che si attivano prendono il nome di neuroni specchio. Solo successivamente si scoprì che anche il cervello umano possedeva questi neuroni specchio, da cui si sono poi susseguite evoluzioni che hanno portato alla nascita della Motor Imagery [14].

### 2.1.3 La corteccia motoria supplementare

La corteccia motoria supplementare (SMA) è coinvolta nella pianificazione di sequenze di movimenti più complessi e nella coordinazione bilaterale. Inoltre la corteccia motoria supplementare sembra essere coinvolta nei movimenti che ricordiamo di aver compiuto.

## 2.2 Ritmi cerebrali

L'attività ritmica cerebrale, come già ampiamente discusso, può essere registrata attraverso sistemi di acquisizione basati su radiazioni elettriche. La scoperta di queste oscillazioni cerebrali si deve ad Hans Berger [19], il primo ad acquisire informazioni dal cervello attraverso Elettroencefalogramma. Solo nel seguito degli anni iniziarono a classificare le onde cerebrali in base alla loro frequenza di oscillazione suddividendole in:

- **Onde Delta( $\delta$ )** : Sono caratterizzate da una frequenza che va da 0,5 a 3,9 Hz, misurabili nelle fasi di sonno.
- **Onde Theta( $\theta$ )** : Compresse nelle frequenze 4Hz a 8Hz, queste onde sono molto presenti nei neonati ed hanno un voltaggio medio di  $100\mu V$ .
- **Onde Alfa( $\alpha$ )** : La frequenza di queste onde varia tra gli 8Hz e i 13Hz, si registrano nelle aree occipitali e parietali con un voltaggio medio di circa  $40\mu V$ .
- **Onde Beta( $\beta$ )** : Il ritmo Beta si registra durante la veglia attiva, in fasi di attenzione e iperattività. Caratterizzate dalla banda di frequenza tra i 13Hz e i 30Hz.
- **Onde Gamma( $\gamma$ )** : Registrate nel range dei 30Hz ai 100Hz, corrispondono a livelli estremi di funzionalità cognitiva, il voltaggio è basso e questo le rende più difficili da studiare.

Di particolare interesse e in linea con lo studio proposto da questa tesi sono i ritmi MU

- **Onde MU( $\mu$ )** : Hanno un contenuto frequenziale tra i 7-12Hz ma possono variare tra soggetti diversi, sono onde associate all'attività motoria, di conseguenza registrate nei pressi della corteccia motoria e caratterizzate da ERD/ERS (Event related Desynchronization/synchronization). Denominato anche Sensori Motor Rhythm (SMR), hanno un'ampiezza ridotta, il che comporta l'utilizzo di processi di elaborazione del segnale per poterle discriminare chiaramente rispetto ad altri segnali.

# Capitolo 3

## Software a supporto

Entrando nel focus di questa tesi, incentrata sulle esperienze acquisite nel corso del mio tirocinio al centro di ricerca "E. Piaggio", trovo utile riportare i principali software utilizzati per l'acquisizione del segnale, per il processamento del dato e per la generazione dell'ambiente virtuale nel quale viene immerso il partecipante attraverso VR.

### 3.1 Acquisizione del dato

Acquisire il segnale cerebrale, in particolare dallo scalpo nel caso di BCI-EEG based, risulta essere un compito alquanto complesso per diversi fattori. In primo luogo, vista la necessità di dover acquisire segnali con ampiezza alquanto ridotta, è necessario utilizzare componenti ad alta sensibilità coadiuvate da convertitori A/D a elevata velocità per la mole di dati da trasmettere. Un headset può variare molto nel prezzo, questo è chiaramente correlato al tipo di acquisizione che si vuole effettuare, tenendo conto di canali acquisiti, precisione dell'acquisizione, frequenza di sampling, ed eventuali funzionalità aggiuntive che possono essere utili per registrare altri segnali fisiologici oltre a quelli generati dal cervello, quali ad esempio EDA (ElectroDermal Activity) o ECG (ElectroCardioGraphy). Una caratteristica importante, particolarmente per applicazioni RT (real time), è data dalla qualità della connessione che collega l'headset al computer che effettua la registrazione dei dati ricevuti. Gli headset cablati risultano più ingombranti da indossare, comportano limitazioni nel movimento del partecipante, con conseguente inapplicabilità in alcuni contesti, d'altra parte, la connessione cablata è più robusta rispetto a una connessione wireless, vittima quest'ultima di maggiore jittering con conseguente ritardo nella trasmissione dei dati.

#### 3.1.1 Trasmissione del segnale - DSI-Streamer

Gran parte dei produttori di headset per l'acquisizione del segnale cerebrale forniscono software proprietari per l'interfacciamento della cuffia con il computer. Nel mio caso, avendo utilizzato l'headset DSI-24 della Wearable Sensing, il software messo a disposizione dalla fabbrica produttrice è DSI-Streamer, visibile in figura 3.1.

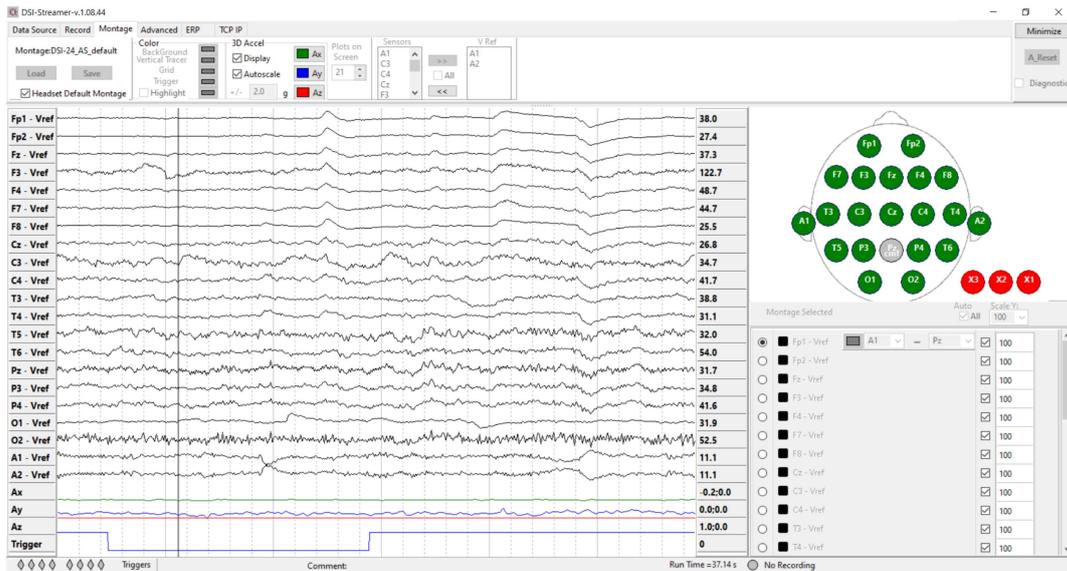


Figura 3.1: In figura vediamo l'interfaccia grafica messa a disposizione dal software DSI-streamer, da questa si evince come, oltre al dato ricevuto e al plot dello stesso in real time, riceviamo informazioni relative alla qualità del segnale acquisito dai singoli elettrodi attraverso dei colori che ci forniscono informazioni sull'affidabilità del dato. Nella colonna di sinistra sono riportati i nomi degli elettrodi oltre al trigger (utile per riportare l'istante di tempo in cui si verifica l'evento d'interesse). Nella colonna di sinistra, sono invece riportati in  $\mu V$  i valori di tensione registrati dai singoli elettrodi.

Il Software consente la connessione alla cuffia attraverso una chiavetta Wi-Fi fornita insieme al prodotto. Mi urge riportare che, essendo le nostre macchine di elaborazione general purpose, è bene "informare" il sistema operativo che la chiavetta in questione dovrà fornire una grande mole di dati nel tempo, è che quindi quest'ultima necessiterà di una banda elevata di trasmissione. In questo senso, attraverso il device manager (Windows), dovremo modificare le impostazioni relative ai bit al secondo trasmessi e al buffer disponibile, i quali altrimenti comporterebbero un collo di bottiglia nello streaming dei dati. Attraverso lo stesso software siamo inoltre in grado di salvare le acquisizioni sotto forma di file CSV o EDF, oppure possiamo effettuare lo stream in real time del dato attraverso un socket che sfrutta il protocollo TCP/IP il quale ci permette di reindirizzare il flusso di dati sia sulla rete locale, sia internamente al nostro computer attraverso l'utilizzo di porte di rete non riservate.

### 3.1.2 Trasmissione del segnale - Lab Streaming Layer

Un altro software a supporto dello streaming dei dati che possiamo utilizzare nelle nostre applicazioni è il LabRecorder che utilizza il protocollo LSL (Lab Streaming Layer) sviluppato dallo Swartz Center for Computational Neuroscience [20] per comunicare con la cuffia. Il suddetto protocollo è supportato da diversi headset e

può essere utilizzato in aggiunta al software fornito dalla casa produttrice. Il modo più semplice per utilizzare questo protocollo è attraverso EEGLab [21] che mette a disposizione alcuni plugin per implementarlo, altrimenti possiamo utilizzare il software LabRecorder che ci fornisce ulteriori funzionalità come la sincronizzazione di diversi streams con diversi frequency sampling rates o la funzionalità di salvataggio del dato in formato XDF. Il flusso di dati può essere inoltre facilmente reindirizzato su diverse macchine con O/S differenti attraverso la rete locale, per garantire il sincronismo vengono inserite delle informazioni relative alla latenza nella trasmissione del dato così che sia possibile utilizzare queste informazioni temporali per sincronizzare i vari streams.

## 3.2 Processamento e analisi del dato - EEGLab

Una volta che il dato è stato acquisito e salvato all'interno del computer, siamo pronti per studiare ciò che abbiamo registrato. Per analizzare l'attività cerebrale è stato utilizzato il software EEGLab [21], un toolbox Matlab per l'analisi di segnali fisiologici quali ad esempio EEG o EMG. Attraverso questo software siamo in grado di applicare diverse tecniche di processamento per estrapolare le informazioni dal RAW data, possiamo difatti utilizzare tecniche come ICA (Indipendet Component Analisis) [22], rimozione degli artefatti, analisi tempo/frequenza e altre, per ricavare le informazioni legate al task che il partecipante deve eseguire. Una descrizione più meticolosa riguardo le funzioni utilizzate nel sistema BCI sviluppato sarà discussa nelle pagine successive.

## 3.3 Ambiente di realtà virtuale - Unity3D

Per lo sviluppo dell'ambiente di realtà virtuale è stato utilizzato il software unity3D, il quale utilizza il motore grafico Unity sviluppato da Unity Technologies. Grazie a quest'ultimo è stato possibile creare un ambiente di realtà virtuale per restituire un feedback visivo al partecipante. Unity supporta diversi linguaggi di programmazione, in particolare

- UnityScript
- C#
- Boo

Il software prevede l'instaurazione di una scena, all'interno della quale sarà possibile poi inserire le risorse (GameObjects) per strutturare l'ambiente visibile all'utilizzatore. Ogni risorsa può essere resa interattiva attraverso uno o più script, chiamati Behaviour, ognuno dei quali caratterizzato dalla stessa struttura e composto principalmente da due metodi come visibile in figura 3.2

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class NewBehaviourScript : MonoBehaviour
6 {
7     // Start is called before the first frame update
8     void Start()
9     {
10         .....
11     }
12
13     // Update is called once per frame
14     void Update()
15     {
16         .....
17     }
18 }
19
```

Figura 3.2: All'interno della classe *MonoBehaviour*, troviamo il metodo *Start()* che rappresenta il punto ideale per l'inizializzazione delle variabili utilizzate all'interno del codice. Il metodo *Update()*, che viene ricorrentemente chiamato per ogni frame update, contiene tutte le funzionalità gestite dai *GameObject* durante l'esecuzione del codice.

# Capitolo 4

## setup di acquisizione

Una volta presa la decisione riguardo l'applicazione che vogliamo sviluppare, e aver appreso con quali metodi è possibile stimolare il partecipante, possiamo procedere nello sviluppo della nostra applicazione. Nel caso particolare affrontato da questa tesi si è sviluppato un sistema in grado di discriminare quale sia l'arto immaginato dal partecipante (mano destra o mano sinistra). Per ottenere questa classificazione è possibile utilizzare strumenti di Machine Learning che vengono addestrati sul dataset di training, durante il quale viene istruito al partecipante il task da effettuare. Per questo motivo, nel corso della sessione di training, al partecipante viene richiesto di osservare un video su un display a 15" posto alla distanza di circa un metro, nel quale vengono mostrate in maniera random le azioni da compiere.

### 4.1 Hardware per l'acquisizione

Per la registrazione dell'attività neuronale è stata utilizzato l'headset DSI-24 della Wearable Sensing, attraverso quest'ultimo siamo in grado di registrare con elevata accuratezza i potenziali che si propagano fino allo scalpo del paziente, e di trasmettere i dati acquisiti attraverso connessione Bluetooth ad una chiavetta proprietaria.



(a) DSI-24 Headset.



(b) Wireless Trigger Hub.

Figura 4.1: *Strumenti Hardware utilizzati per l'acquisizione*

I sensori EEG del sistema DSI-24 sono montati in una cuffia leggera e regolabile dall'utente, mostrata in figura 4.1a, nella quale i sensori si trovano nelle posizioni nominali Fp1, Fp2, F7, F3, Fz, F4, F8, T3, C3, Cz, C4, T4, T5, P3, P4, T6, O1 e O2

del sistema internazionale 10/20. L'headset ha inoltre a disposizione il sensore di riferimento (Common Mode Follower, CMF) collocato nella posizione nominale Pz. La cuffia supporta ulteriori 3 canali per lo stream di dati che possono essere utilizzati a piacimento dall'utilizzatore, nel nostro caso questi non sono stati utilizzati. La frequenza massima di sampling rate è di 300Hz. Per sincronizzare il dato EEG con gli eventi è stato utilizzato il trigger Hub della Wearable Sensing, mostrato in figura 4.1b, al quale è stato collegato un fotodiodo che, una volta posizionato sullo schermo, registra gli istanti di tempo in cui occorrono gli eventi.

## 4.2 Video di training

Il video per l'acquisizione del dataset di training è stato sviluppato attraverso del codice in Matlab che ci permette di generare in maniera casuale l'occorrere degli eventi. In figura 4.2 è riportata una rappresentazione schematica di come quest'ultimo è composto.

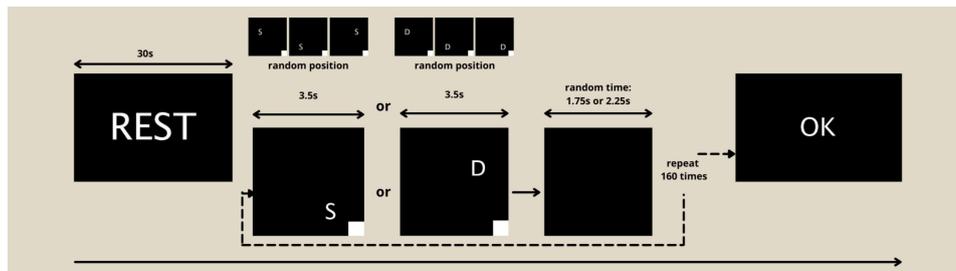


Figura 4.2: Come si osserva, dopo un periodo di REST, introdotto per permettere al paziente di concentrarsi, si susseguono in maniera casuale delle immagini contenenti la lettera S o D, seguite in entrambi i casi da una immagine nera. Durante il display di questa immagine nera viene chiesto al paziente di non pensare al movimento dell'arto. Al presentarsi della lettera S, sarà richiesto al partecipante d'immaginare il movimento della mano sinistra e, analogamente, al presentarsi della lettera D, sarà richiesto al partecipante d'immaginare il movimento della mano destra. Al termine del video si presenta un immagine "OK" che informa il partecipante del termine della sessione di training. Ogni immagine S o D ha un piccolo riquadro bianco nell'angolo in basso a destra, il quale innesca il fotodiodo che ci permette di registrare con precisione l'istante in cui occorre l'evento.

Il video in questione ha subito alcune evoluzioni nella fase di sviluppo, in particolare è stata introdotta un'altra fase di REST nel mezzo del video per consentire al partecipante di muoversi senza invalidare le epoche acquisite con artefatti muscolari. Un'altra modifica è stata fatta nella casualità di posizionamento delle lettere, questo per rimuovere la correlazione con il posizionamento degli occhi tra i trial. Infine è stato modificato il tempo di permanenza dell'immagine nera seguente alle lettere-evento, che viene generato in maniera casuale, così da distruggere il pattern temporale che si crea con la sequenzialità delle immagini.

# Capitolo 5

## Preparazione e modifica del dato

Una volta completata l'acquisizione del dato, attraverso l'utilizzo di metodi e tecniche illustrate nel capitolo precedente, siamo in grado di analizzare la registrazione partendo dal file CSV. Il software che utilizziamo in questa analisi è il già menzionato EEGLab [21], che mette a disposizione dell'utilizzatore diverse tecniche di processamento e visualizzazione del segnale elettroencefalografico.

### 5.1 Caricamento e gestione delle acquisizioni

L'importazione del dato all'interno di EEGLab potrebbe non essere consistente con quanto registrato, questo per via del fatto che non esiste un unico standard di acquisizione e di salvataggio dei file, comportando alcune ambiguità che ricadono in un'incorretta visualizzazione dei valori. Per questo motivo è sempre bene analizzare il file in maniera approfondita attraverso editor di testo quali ad esempio NotePad++, per esser certi che il file sia stato formattato nella maniera corretta. Una volta superato questo step, troviamo le nostre acquisizioni caricate all'interno di EEGLab, e possiamo procedere con l'analisi del segnale per verificare che tutto sia stato correttamente registrato durante l'acquisizione. Si procede in questo senso alla rappresentazione grafica dei valori attraverso la visualizzazione del tracciato EEG, per far questo EEGLab mette a disposizione la funzionalità di plot che, una volta definita la locazione dei canali, fornisce un grafico come quello riportato in figura 8.1. Nel caso particolare di BCI Event-based, è necessario importare gli eventi all'interno del tracciato EEG per poter in un secondo momento informare il classificatore supervisionato quando questi si presentano. In questo caso, avendo acquisito l'evento di trigger direttamente dal Software DSI-streamer attraverso l'applicazione del fotodiode, conosciamo i tempi di latenza di questi e ci basta quindi marcarli per sapere quale si riferisce all'evento di "mano destra" piuttosto che di "mano sinistra". Per fare questo EEGLab ci permette d'importare le informazioni relative agli eventi direttamente da file ASCII o variabili Matlab, o di modificarli manualmente. Una volta quindi importata la sequenzialità con la quale occorrono gli eventi nel corso del video proposto al partecipante, il tracciato si arricchirà d'informazioni relative agli eventi, evidenziando in particolare il tipo d'evento (destra o sinistra) e l'istante temporale in cui questi si verificano.

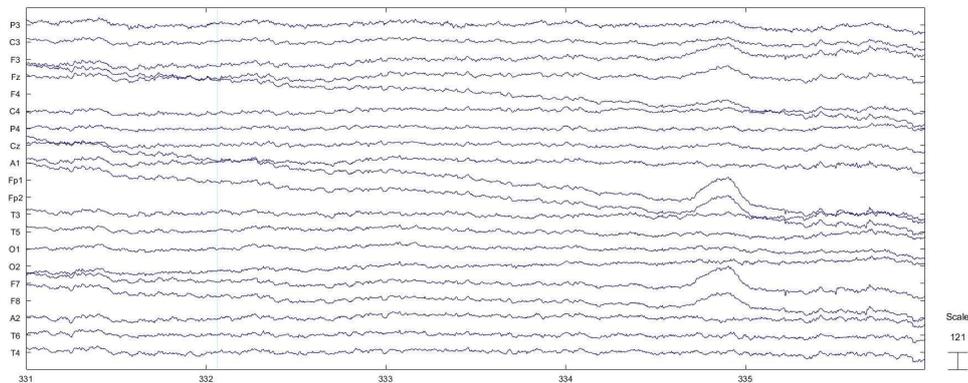


Figura 5.1: Si osserva il plot dei dati per ogni canale, tale funzionalità è raggiungibile attraverso il menu di EEGLab attraverso Plot  $\rightarrow$  Channel data (scroll)

## 5.2 Filtraggio del dato

Le operazioni di filtraggio messe a disposizione da EEGLab invalidano a tutti gli effetti il dataset di training per il classificatore, questo perchè in attività Online non sarà possibile applicare lo stesso tipo di filtraggio in real-time allo stream di dati. In ogni caso, utilizzare tecniche di filtraggio del segnale ci permettono di effettuare un'analisi più meticolosa, così da identificare la finestra di tempo e le bande di frequenza (che variano di partecipante in partecipante) soggette all'occorrere dell'evento. Per risalire a queste informazioni in primo luogo è bene filtrare il segnale in frequenza per pulirlo da artefatti muscolari, oculari, o da altre attività cerebrali che si manifestano nel corso dell'acquisizione, così come il rumore introdotto dalla rete di alimentazione a 50Hz. Applichiamo perciò un filtro FIR passa-banda 8-30Hz e procediamo all'estrazione delle epoche attraverso la funzionalità di EEGLab che ci permette di dividere in due il dataset originale, dove ognuno di questi conterrà eventi di un solo tipo. Possiamo ora identificare, attraverso l'analisi del grafico tempo-frequenza, i parametri che possiamo fornire al CSP per applicare la trasformazione. In figura 5.2 è riportato il grafico dal quale deduciamo la finestra di tempo e le bande di frequenza.

## 5.3 Pulizia del tracciato EEG

Come affermato nel paragrafo precedente, il Dato filtrato non è valido per il training di un classificatore in real-time, vista la sua inconsistenza con un possibile stream non filtrato. Per questo, una volta determinati i parametri d'input per il Filtraggio spaziale attraverso CSP, e averne analizzato la sua validità in termini di acquisizione, ripartiamo dal dato originale non filtrato e operiamo attraverso le

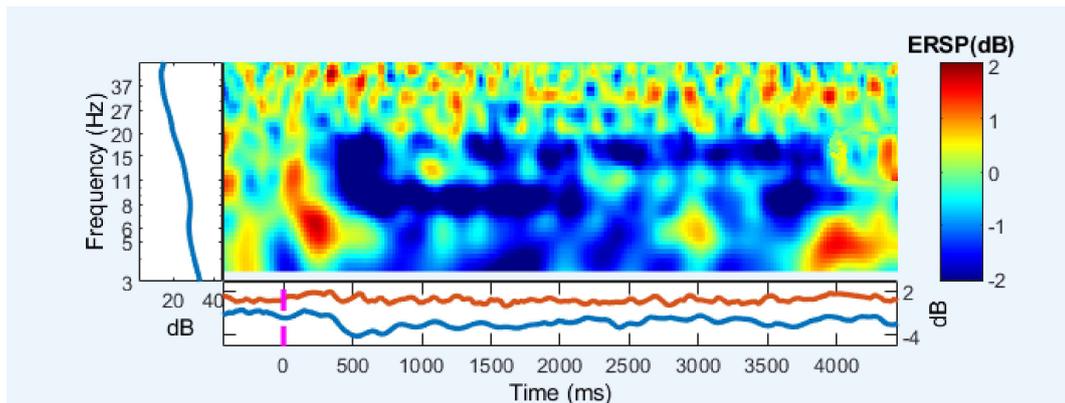


Figura 5.2: Grafico tempo-frequenza del canale C3 ricavato dalle funzionalità di EEGLab, si evince nel mio particolare caso che la finestra di tempo significativa inizia circa 300ms dopo l'occorrenza dell'evento e termina poco dopo il termine dell'evento, le bande di frequenza sono quelle Alpha e Beta, con un maggiore contenuto informativo, nel mio caso, nelle onde Alpha. Le aree di colore Blu identificano una soppressione di potenza, in linea con quanto affermato riguardo la desincronizzazione (ERD) che si verifica all'occorrere dell'evento.

possibilità fornite da EEGLab per rimuovere segmenti del segnale ricchi di artefatti che possono peggiorare le performance del classificatore.

### 5.3.1 Rimozione dei segmenti compromessi

EEGLab fornisce sistemi automatici o manuali di rimozione dal tracciato EEG di segmenti compromessi, benché il sistema automatico operi con buona validità, la rimozione manuale garantisce risultati migliori, che si traducono in un incremento di performance del classificatore. Il tracciato EEG è spesso caratterizzato da trend ricorrenti relativi ad artefatti che possono essere identificati facilmente, alcuni di questi sono riportati di seguito.

- **Artefatti Muscolari:** Si presentano a causa dei movimenti del partecipante durante l'acquisizione, hanno frequenze variabili a seconda della grandezza del muscolo e dal rateo d'innervatura dello stesso, l'attività si manifesta nella banda di frequenza che va dai 15Hz fino ai 200/300Hz con picchi che possono raggiungere i 600Hz per particolari artefatti muscolari facciali[23]. Sono in generale facilmente riconoscibili all'interno del tracciato EEG.
- **Artefatti Oculari:** L'attività oculare genera forti oscillazioni all'interno del tracciato EEG, in particolare i blink, causati dalla differenza di potenziale presente tra cornea e il fondo dell'occhio, sono equivalenti a un dipolo elettrico orientato verso la cornea, e si generano nel momento in cui la palpebra scivola verso il basso sulla cornea.

- **Artefatti Cardiaci:** Gli artefatti risultanti dall'attività elettrica cardiaca contaminano il tracciato EEG con picchi di potenza che possono oscurare il contenuto informativo. Attraverso sistemi ibridi di acquisizione siamo in grado di rimuovere questi artefatti anche in situazioni Real-time [24]

Oltre agli artefatti fin'ora elencati, che possono essere categorizzati come artefatti fisiologici in quanto relativi alle funzioni organiche del partecipante, Gli artefatti non fisiologici sono causa di una incorretta applicazione degli elettrodi o di un malfunzionamento del sistema il che comporta la registrazione d'informazioni non consistenti con la reale attività cerebrale. Una strada che possiamo prendere per ovviare all'incorretta applicazione di un elettrodo sullo scalpo del partecipante è attraverso l'interpolazione dei canali, funzionalità presente su EEGLab, la quale permette di restaurare il segnale acquisito da un canale a partire dall'attività elettrica registrata dagli elettrodi adiacenti. Dopo aver quindi preparato il dataset di training, secondo le operazioni di filtraggio e di pulizia del tracciato come trattato in questo capitolo, possiamo attraverso il software BCILab procedere con il training del classificatore.

## Capitolo 6

# BCILAB, pre-processing e classificazione

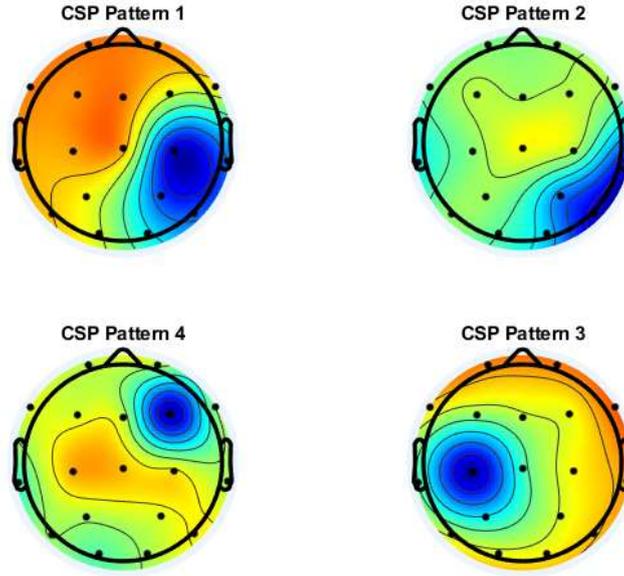
Prima d'introdurre il software BCILab, il quale ci permette di adottare tecniche di machine learning per l'apprendimento del classificatore, mi urge introdurre da un punto di vista teorico e matematico, il funzionamento degli algoritmi utilizzati per il training del classificatore.

### 6.1 CSP - Common Spatial Pattern

Il CSP (Common Spatial Pattern) è uno degli algoritmi più popolari nel campo delle BCI, nello specifico per processi oscillatori come la Motor Imagery, Benjamin et al.[25]. Come già discusso in precedenza, i potenziali acquisiti dall'elettroencefalogramma hanno una bassa risoluzione spaziale per via della dispersione del segnale nel mezzo conduttore, per questo motivo, segnali deboli come l'SMR (SensoriMotor rhythms) sono difficili da distinguere e studiare in quanto altri segnali possono produrre potenziali più elevati nelle stesse bande di frequenza. Il CSP diventa funzionale per ovviare a questa problematica, quest'ultimo difatti applica una trasformazione lineare data dalla matrice  $W$  per proiettare il dato EEG multicanale in uno spazio dimensionale più piccolo dove le sorgenti sono un mix dei canali originali.

$$Y = WX$$

$Y$  è una matrice (Source x Samples) mentre  $X$  è la matrice del dato originale (Channels x Samples). Ogni riga della matrice  $W$  è composta da termini di peso per ogni canale, mentre le colonne della matrice  $(W^{-1})^T$  sono chiamati spatial patterns. Prima di applicare questa trasformazione spaziale, è bene identificare sia la finestra di tempo all'interno della quale si verifica lo stimolo, sia la banda di frequenza coinvolta nella desincronizzazione (ERD). Queste bande di frequenza possono cambiare tra un soggetto e un altro, in genere può essere utile osservare, come anticipato in precedenza nel capitolo 5, il diagramma tempo-frequenza dei canali più interessati dall'evento, come ad esempio gli elettrodi C3 o C4, per selezionare i giusti parametri relativi a banda di frequenza e finestra di tempo.



Modello ottenuto applicando l'algoritmo CSP sul mio Dataset pre-processato, rappresentano il tracciamento delle prime due e ultime due colonne della matrice  $(W^{-1})^T$ . I pattern 1 e 3 sono i più informativi.

### 6.1.1 Procedure matematiche

Assumiamo di avere  $N$  trial per ogni classe di evento, le matrici  $X_{L,1}$  e  $X_{R,1}$  rappresentano il dato acquisito dall'EEG nel primo trial per entrambe le classi destra e sinistra.  $X_{L,1}$  e  $X_{R,1}$  hanno una dimensione (Num Channels X Samples in un trial).

La matrice di covarianza normalizzata per il primo trial destro e per il primo trial sinistro sono calcolate come in Eq. 6.1

$$C_{L,1} = \frac{X_{L,1}X_{L,1}^T}{Trace(X_{L,1}X_{L,1}^T)} \quad C_{R,1} = \frac{X_{R,1}X_{R,1}^T}{Trace(X_{R,1}X_{R,1}^T)} \quad (6.1)$$

$X^T$  è la matrice trasposta di  $X$  mentre  $Trace(\cdot)$  è definita come la somma di tutti gli elementi sulla diagonale. Dopo aver calcolato la matrice di covarianza normalizzata per ogni trial, sia per classe destra che sinistra, la matrice di covarianza media per le due classi è calcolata come segue:

$$\overline{C}_L = \frac{1}{N} \sum_{k=1}^N C_{L,k} \quad \overline{C}_R = \frac{1}{N} \sum_{k=1}^N C_{R,k} \quad (6.2)$$

La somma delle matrici  $\overline{C}_L$  e  $\overline{C}_R$  può essere decomposta secondo la scomposizione

in autovalori e autovettori come segue:

$$C = \overline{C}_L + \overline{C}_R = U_0 \Sigma U_0^T$$

dove  $U_0$  sono gli autovettori e  $\Sigma$  è una matrice diagonale composta dagli autovalori legati alla matrice  $C$ . Tale decomposizione è permessa in quanto  $C$  è una matrice reale e simmetrica.

Gli autovettori sono reali e ortonormali, in questo modo abbiamo che  $U_0^T = U_0^{-1}$ .

Di seguito calcoliamo la matrice di sbiancamento  $P$

$$P = \sqrt{(\Sigma)^{-1}} U_0^T$$

La matrice di sbiancamento (Matrice di whitening) è una trasformazione lineare che trasforma un vettore di variabili con matrice di covarianza nota in un nuovo insieme di variabili la cui matrice di covarianza è la matrice identità. Applicando questa trasformazione a entrambe le matrici  $\overline{C}_L$  e  $\overline{C}_R$  otteniamo:

$$S_L = P \overline{C}_L P^T \quad S_R = P \overline{C}_R P^T$$

La matrice  $S_L$  e  $S_R$  possono essere ulteriormente decomposte come segue:

$$S_L = V \Sigma_L V^T \quad S_R = V \Sigma_R V^T$$

$S_L$  e  $S_R$  hanno gli stessi autovettori  $V$ , e la combinazione dei corrispettivi autovalori ( $\Sigma_L$  and  $\Sigma_R$ ) per entrambe le classi è uguale a uno. in altre parole possiamo scrivere  $\Sigma_L + \Sigma_R = I$  dove  $I$  è la matrice identità. Difatti, se gli autovalori di  $S_L$  hanno un valore più prossimo ad 1, gli stessi autovalori di  $S_R$  hanno un valore prossimo a 0, e viceversa.

Infine, la matrice di proiezione  $W$  è data da:

$$W = V^T P$$

## 6.2 Un introduzione ai classificatori

Il modello di regressione logistica, l'analisi lineare discriminante (LDA) o il Support Vector Machine (SVM) sono solo alcuni dei classificatori più recentemente implementati per la classificazione di misure o casi "Nuovi" rispetto a quelli utilizzati per stimare i parametri del modello. I classificatori cercano d'indovinare l'appartenenza di un elemento, un soggetto o un set di misure a una classe. I Dataset sono collezioni di dati che utilizziamo per addestrare i nostri classificatori, in particolare i Dataset possono essere suddivisi a secondo del loro utilizzo:

- Dataset di training: Dati sui quali vengono stimati i parametri del modello, ogni dato è legato a un etichetta che ne indica la vera classe di appartenenza.

- Dataset di Validazione: Serve per decidere alcuni iper-parametri che non possono essere decisi dai dati (Es: la struttura del modello usato per la classificazione).
- Dataset di test: Su questo si determina la performance del classificatore.

### 6.2.1 Iper-parametri

Spesso gli iper-parametri vengono definiti come quei parametri che vengono decisi prima della stima dei parametri. Ad esempio, nella regressione logistica questi potrebbero essere:

- I valori iniziali
- La scelta dell'approccio o algoritmo per la stima dei parametri
- Il numero d'iterazioni massimo

In questo caso verranno quindi stimati diversi modelli utilizzando diversi iper-parametri. La stima sarà effettuata sul dataset di training e poi la decisione sui migliori iper-parametri verrà effettuata in base ai risultati sul dataset di validazione. In diversi casi, la terminologia è contraddittoria e non omogenea. Va detto però che esistono delle strategie per utilizzare un dataset sia come validation che come training suddividendolo in modo opportuno in diversi sottoinsiemi che saranno utilizzati alternamente come dataset di training o di validazione, questa tecnica prende il nome di Cross-Validation. Un ulteriore parametro che dovrà essere stimato è proprio la soglia di cut-off, che sarà applicata alle stime per discriminare se un caso (osservazione) appartenga o meno a una classe o all'altra.

### 6.2.2 Prestazione dei classificatori

Prima di esporre le diverse tipologie di classificatore, vediamo alcune misure che permettono di valutare le prestazioni di un classificatore. Consideriamo a tal proposito le classi A e B relativamente come casi negativi o casi positivi. All'atto della classificazione è possibile distinguere 4 casi:

In analogia ai risultati del test sulle ipotesi si parla:

- Errori di I specie, quando un caso è negativo e viene considerato positivo (FP)
- Errori di II specie, quando un caso è positivo e viene classificato come negativo (FN)

### 6.2.3 Sensibilità e Specificità

Nella valutazione delle performance di un classificatore si utilizzano due indici che sono determinabili a partire dalla matrice di confusione.

	<b>Predicted</b>		
		<b>Positive</b>	<b>Negative</b>
<b>Actual</b>	<b>Positive</b>	True Positive (TP)	False Negative (FN)
	<b>Negative</b>	False Positive (FP)	True Negative (TN)

Figura 6.2: Tabella di contingenza o Matrice di confusione per classificatori binari

La sensibilità di un classificatore è definita come la capacità di classificare correttamente casi/osservazioni della classe A, è definita come la frazione di soggetti correttamente identificati come positivi.

$$\text{Sensibilità o TPR} = \frac{VP}{\text{TotalePositivi}} = \frac{VP}{VP + FN} \quad (6.3)$$

La specificità di un classificatore è definita come la capacità di classificare correttamente i casi/osservazioni appartenenti alla classe B, è definita come la frazione di soggetti correttamente identificati come negativi.

$$\text{Specificità o TNR} = \frac{VN}{\text{TotaleNegativi}} = \frac{VN}{VN + FP} \quad (6.4)$$

Come già menzionato, la scelta di una soglia per classificatori binari rappresenta il parametro fondamentale per discriminare casi/osservazioni. La modifica di questa soglia modifica i valori della matrice di confusione e quindi i valori di specificità e sensibilità, Un modo per studiare la variazione di questi parametri in funzione della soglia è l'analisi delle curve ROC (Receiver Operating Characteristic).

#### 6.2.4 Le curve ROC

Le curve ROC sono un metodo grafico 6.3 che mette in relazione sensibilità e specificità al variare del valore di cut-off. Gli indici di bontà comunemente utilizzati sono due:

- TPR: True Positive Rate o Sensibilità
- FPR: False Positive Rate, errori commessi nel non riconoscere i negativi o falsi allarmi. calcolabile come  $1 - \text{Specificità}$

I falsi allarmi (FPR) salgono all'aumentare del true positive rate, il tentativo di migliorare la capacità di classificare i veri positivi si traduce in un aumento di

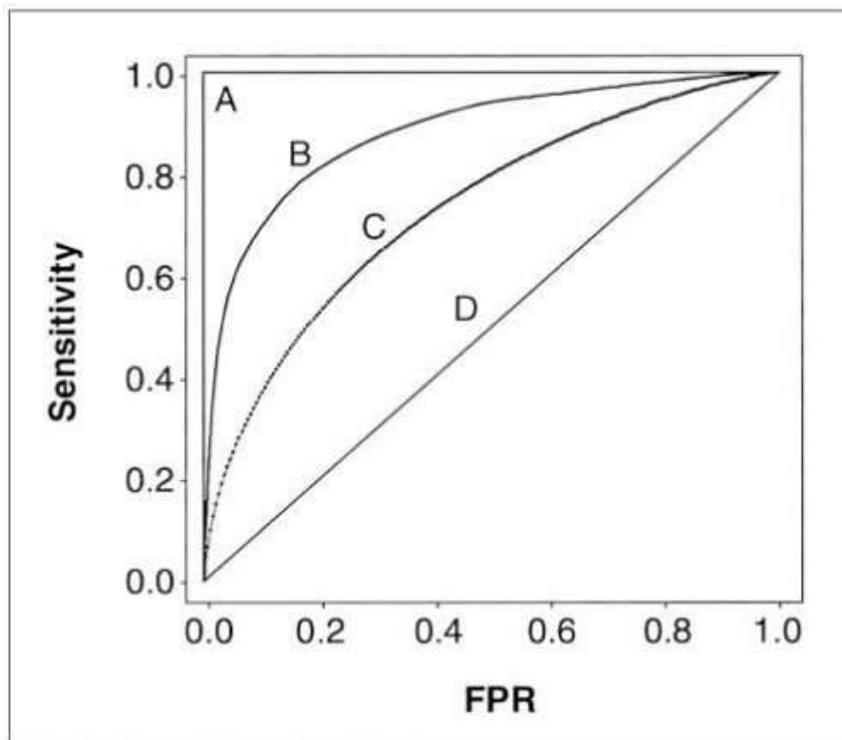


Figura 6.3: Ogni curva è relativa a un diverso valore di cut-off per il classificatore, La curva A (ideale) rappresenta gli indici di bontà di un classificatore reale, al contrario, la curva D rappresenta un classificatore casuale. In breve, più ci allontaniamo dalla diagonale, migliori saranno le prestazioni del classificatore.

falsi positivi ovvero riduzione della specificità. volendo analizzare le curve ROC per scegliere la soglia statistica "ottima", procediamo con la costruzione di una funzione di costo di errata classificazione, questa funzione di costo dipenderà dal tipo di applicazione e in particolare il costo che diamo a un errore di FP piuttosto di FN può variare.

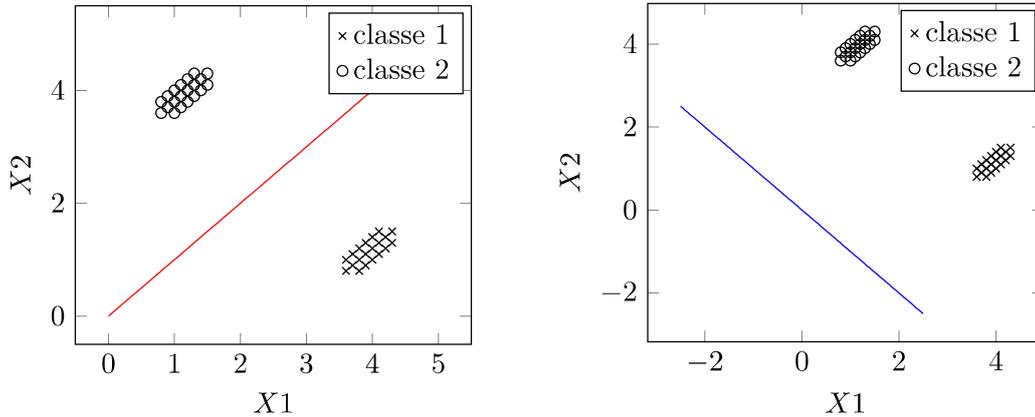
$$\text{Costo} = \text{FP} \cdot \text{CFP} + \text{FN} \cdot \text{CFN} \quad (6.5)$$

dove CFP e CFN rappresentano i costi di un falso positivo e di un falso negativo, rispettivamente. Se i costi di CFP e CFN si equivalgono, possiamo scegliere la soglia direttamente in funzione di sensibilità e specificità.

### 6.3 LDA - Linear Discriminant Analysis

L'Analisi Discriminante Lineare (LDA) rappresenta uno degli algoritmi di apprendimento più classici, originariamente introdotto da Fisher [26] e tutt'ora utilizzato in diversi ambiti come il riconoscimento facciale [27] e l'immagine retrieval [28].

L'LDA, date due classi (o più classi attraverso il modello 1-vs-1), a cui appartengono due distribuzioni normali, determina una direzione di proiezione dei dati in modo da



(a) *Caso Peggior*: La proiezione delle misure sulla retta rende impossibile trovare un valore di soglia per distinguere l'appartenenza delle misure alle due classi.

(b) *Caso Migliore*: La proiezione delle misure sulla retta rende semplice trovare un valore di soglia per distinguere l'appartenenza delle misure alle due classi.

Figura 6.4: *Caso Bidimensionale, 2 classi*

ottimizzare la loro distinzione.

Essendo tale metodo supervisionato, la direzione di proiezione dei punti  $X_i$  può essere determinata solo se si conoscono le classi delle misure, delle quali si assume che la varianza delle misure sia la stessa per ogni classe. Una volta effettuata la trasformazione, si identificherà un valore di cut-off che discriminerà l'appartenenza delle successive misure a una classe. Si consideri il caso di 2 variabili tale che le

$$\text{misure } X_i = \begin{bmatrix} x_{1i} \\ x_{2i} \end{bmatrix}$$

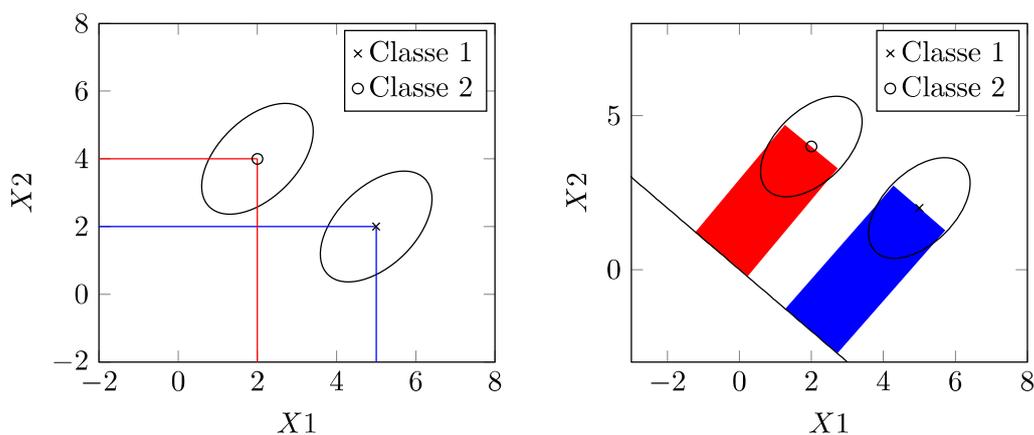
Trovandoci in uno spazio di dimensione 2 siamo in grado di riportare graficamente le distribuzioni per le due classi come fatto in Fig: 6.4. Dovendo determinare l'iperpiano di proiezione, essendo il caso bidimensionale, dobbiamo trovare una retta sulla quale proiettare le misure di ogni classe.

Consideriamo ora il caso più generale in cui vi siano P variabili t.c.  $X_i = \begin{bmatrix} x_{1i} \\ \vdots \\ x_{Pi} \end{bmatrix}$

Ciò che vogliamo determinare sono i vettori  $a^T$  tale per cui  $d = a^T \cdot X_i$  dove  $d$  è la distanza dall'origine della proiezione di  $X_i$

Un modo per trovare la direzione  $a^T$  è quella di massimizzare la distanza tra le medie delle classi e nel frattempo minimizzare la dispersione delle proiezioni in ogni classe. Per far ciò, partiamo trovando le medie di partenza per ogni classe:

$$\mu_x = \frac{1}{N_x} \sum_{X_k \in W_x} X_k \quad (6.6)$$



(a) Le due proiezioni lungo gli assi  $X$  e  $Y$ , Massimizzare la differenza  $|\tilde{\mu}_1 - \tilde{\mu}_2|$  non è sufficiente. (b) La proiezione tiene conto della dispersione dentro ogni classe (Within class variability).

Figura 6.5: Proiezioni delle classi su diversi assi, caso Bidimensionale

La quale media, una volta effettuata la proiezione diviene:

$$\tilde{\mu}_x = \frac{1}{N_x} \sum_{X_k \in W_x} a^T X_k = a^T \frac{1}{N_x} \sum_{X_k \in W_x} X_k \quad (6.7)$$

nell'ipotesi di due classi,  $W_1$  e  $W_2$ , avremo:

$$\begin{aligned} \tilde{\mu}_1 &= a^T \mu_1 \\ \tilde{\mu}_2 &= a^T \mu_2 \end{aligned} \quad (6.8)$$

da ciò si evince che dobbiamo massimizzare  $|\tilde{\mu}_1 - \tilde{\mu}_2|$

Oltre a massimizzare la differenza tra la media delle due classi, dobbiamo garantire che la proiezione minimizzi la dispersione delle misure all'interno di ogni classe, un esempio geometrico di questo è riportato in figura 6.5

La dispersione Sappiamo essere definita, per una classe  $k$ , come:

$$\tilde{S}_k^2 = \sum_{Y \in W_k} (Y - \tilde{\mu}_k)^2 \quad (6.9)$$

Fisher, per cui L'LDA si chiama anche FDA, determina questa funzione di costo per due classi, che punta a massimizzare il termine  $J(a)$ :

$$J(a) = \frac{|\tilde{\mu}_1 - \tilde{\mu}_2|}{\tilde{S}_1^2 + \tilde{S}_2^2} \quad (6.10)$$

Dobbiamo quindi determinare i coefficienti di  $a = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix}$  t.c.  $J(a)$  sia massima

Definiamo la matrice di dispersione within class:

$$S_k = \sum_{x \in W_k} (x - \mu_k)(x - \mu_k)^T \quad (6.11)$$

Nel caso di due classi avremo quindi  $S_1$  e  $S_2$  t.c.  $S_w = S_1 + S_2$ . Il vettore di proiezione a proietta ogni  $x_i$  t.c

$$y_i = a^T x_i \quad \forall i \quad (6.12)$$

Di conseguenza la matrice di dispersione  $\tilde{S}_k$  può essere espressa in termini di del vettore di proiezione a:

$$\begin{aligned} \tilde{S}_k &= \sum_{y \in W_k} (y - \tilde{\mu}_k)(y - \tilde{\mu}_k)^T = \sum_{x \in W_k} (a^T x - a^T \mu_k)(a^T x - a^T \mu_k)^T = \\ &= a^T \left( \sum_{x \in W_k} (x - \mu_k)(x - \mu_k)^T \right) a = a^T S_k a \end{aligned} \quad (6.13)$$

Per due classi:

$$\tilde{S}_1 + \tilde{S}_2 = a^T S_1 a + a^T S_2 a = a^T S_w a \triangleq \tilde{S}_w \quad (6.14)$$

Analogamente facciamo con le medie:

$$(\tilde{\mu}_1 - \tilde{\mu}_2)^2 = (\tilde{\mu}_1 - \tilde{\mu}_2)(\tilde{\mu}_1 - \tilde{\mu}_2)^T \quad (6.15)$$

Per cui, essendo  $S_B = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$  dove  $S_B$  è la dispersione tra classi (Between class)

$$(\tilde{\mu}_1 - \tilde{\mu}_2)^2 = a^T S_B a \triangleq \tilde{S}_B \quad (6.16)$$

Riprendendo la funzione di costo  $J(a)$ , e volendone trovare il massimo facciamo la derivata e la poniamo a 0

$$\frac{d}{da} [J(a)] = \frac{d}{da} \left[ \frac{a^T S_B a}{a^T S_w a} \right] = \frac{a^T S_w a \frac{d}{da} [a^T S_B a] - a^T S_B a \frac{d}{da} [a^T S_w a]}{(a^T S_w a)^2} = 0 \quad (6.17)$$

A questo punto, dopo la serie di calcoli, otteniamo:

$$S_w^{-1} S_B a = J(a) a \quad (6.18)$$

infine, ponendo  $a = S_w^{-1}(\mu_1 - \mu_2)$  si trova la soluzione.

A questo punto basta scegliere la soglia  $c$ :  $\begin{cases} a^T x > c \longrightarrow W_1 & (a^T x + a_0 > 0 \longrightarrow W_1) \\ a^T x < c \longrightarrow W_2 & (a^T x + a_0 < 0 \longrightarrow W_2) \end{cases}$

Se si considerano uguali gli errori di I e II specie, si sceglie  $c = \frac{(\tilde{\mu}_1 + \tilde{\mu}_2)}{2}$

## 6.4 BCILAB

BCILAB è il software a supporto che utilizziamo per allenare il classificatore, creato e sviluppato da Christian Kothe [29], ci permette, attraverso una semplice interfaccia grafica, di effettuare analisi online e offline sul dataset di training. In particolare, una volta caricato il dataset, possiamo definire un particolare approccio che sarà utilizzato per fare il training del classificatore. Come anticipato, l'approccio utilizzato sarà composto dal filtro spaziale CSP per l'estrazione delle features e la riduzione delle sorgenti, il sottospazio risultante sarà poi analizzato dall'algoritmo LDA per determinare un iperpiano di suddivisione che permetta la discriminazione dell'evento immaginato dal partecipante. Una volta definiti i parametri utili al CSP per applicare il filtraggio spaziale, quali la banda di frequenza e la finestra di tempo, possiamo definire il numero di pattern da estrapolare, nel mio caso questi saranno 2 per classe.

### 6.4.1 Analisi Offline

Nel corso dell'analisi offline, quello che facciamo una volta allenato il nostro classificatore, e di testare le sue performance. In primo luogo andiamo ad applicare la tecnica di cross-validation che prevede la suddivisione del dataset di training in diverse parti, alcune di queste vengono utilizzate per allenare il classificatore mentre altre vengono utilizzate come test per quest'ultimo. I valori di miss-classification raggiunti da questo tipo di approccio variano tra il 75% e il 90%, seppur questo valore varia a seconda della qualità dell'acquisizione e dal soggetto in esame. Un altro modo che abbiamo per testare la robustezza del nostro classificatore e quella di utilizzare un ulteriore dataset, il dataset di Validation, sul quale essendo sempre presenti i label che descrivono la reale appartenenza di un evento a una suddetta classe, è in grado di fornirci il valore di performance. È stato osservato che l'utilizzo di diversi dataset per la valutazione delle performance del classificatore forniscono risultati molto differenti, questa alta variabilità è attribuibile a una differente applicazione della cuffia sullo scalpo del partecipante tra le due acquisizioni. È bene infatti acquisire il segnale cerebrale del partecipante in un'unica sessione, così da non alterare il posizionamento degli elettrodi.

### 6.4.2 Analisi Online

Nel corso dell'analisi online, è possibile effettuare lo stream dei dati attraverso l'headset, in questo modo il classificatore potrà discriminare in real-time il movimento della mano destra o sinistra in maniera asincrona, cioè senza effettivamente conoscere il momento in cui il partecipante inizierà a immaginare il movimento. attraverso il plugin fornito da DSI-Streamer, siamo in grado di fornire in ingresso al classificatore il dato in real-time, infine ci è possibile, attraverso un socket TCP/IP, l'inoltro dell'output del classificatore. Nel nostro caso, il dato sarà inoltrato al software Unity

al fine di avere una compatibilità anche con il sistema di realtà virtuale che fornirà il feedback visivo al partecipante.



# Capitolo 7

## Ambiente di realtà virtuale

La realtà virtuale nasce dalla combinazione di dispositivi hardware e software che cooperano al fine di creare un ambiente che simuli la realtà. Utilizzata sempre in più ambiti, la realtà virtuale (VR) riscuote il suo successo grazie ai recenti avanzamenti in campo informatico ed elettronico, che hanno permesso la costruzione di ambienti sempre più articolati e complessi. Benchè la realtà virtuale sia riconosciuta dalla massa come piattaforma video ludica, le sue applicazioni spaziano dal settore manageriale a quello medico, fornendo uno strumento di apprendimento e di lavoro che mira a sostituire la realtà. Nel 1968 Ivan Sutherland [30], creò quello che è considerato il primo sistema di realtà virtuale con visore, il quale risultava ingombrante e poco funzionale, in quanto utilizzava sistemi meccanici rispetto alla sensoristica presente al giorno d'oggi. Per utilizzare un sistema di realtà virtuale sono generalmente necessari tre dispositivi hardware:

- Un visore con display che avvolge la vista del partecipante.
- Un computer con elevata capacità di calcolo.
- Un sistema d'input per l'interfacciamento dell'utente con l'ambiente simulato che lo circonda, in generale composto da una coppia di joypad.

Nel particolare di questa tesi, possiamo caratterizzare i principali sistemi utilizzati per lo sviluppo dell'ambiente di realtà virtuale. Il visore utilizzato è l'oculus rift S, mentre il sistema d'input originale, composto da due joypad, è stato sostituito dall'output del classificatore che discrimina i movimenti della mano destra o sinistra in tempo reale.

### 7.1 Unity3D

Già discusso come software a supporto nel capitolo 3, unity3D è stato utilizzato per la costruzione dell'ambiente di realtà virtuale grazie alla sua compatibilità con diversi visori VR. Il linguaggio di programmazione che ho utilizzato per gestire sia il socket TCP/IP che le animazioni è C#.

### 7.1.1 Socket TCP/IP

Il primo passo è quello di trovare il modo con il quale Unity può ricevere il dato dal classificatore, essendo entrambi i software presenti all'interno dello stesso computer sfruttiamo il protocollo TCP/IP per instaurare una connessione logica interna alla macchina per lo scambio d'informazione tra i due processi. In particolare il software BCILAB fungerà da server per l'invio dei dati, avendo al suo interno integrato un sistema di socket server TCP/IP, mentre è necessaria qualche riga di codice in Unity per creare il client socket. Il TCP/IP è un protocollo di rete a pacchetto al livello di trasporto, il quale si occupa del collegamento dei due processi attraverso l'utilizzo di porte di rete non riservate. Il codice utilizzato fa uso di librerie che permettono la facile implementazione del protocollo in C#, in particolare la libreria utilizzata è Net.Sockets.

```
void TCP()
{
    server = null;
    try
    {
        Int32 port = 12346;
        IPAddress localAddr = IPAddress.Parse("127.0.0.1");
        server = new TcpListener(IPAddress.Any, port);

        // Start listening for client requests.
        server.Start();

        // Buffer for reading data
        Byte[] bytes = new Byte[256];
        String data = null;
```

Come si osserva in questa prima sezione di codice, si vanno a inizializzare le informazioni relative al collegamento, in particolare la porta e l'indirizzo IP 127.0.0.1 che corrisponde al Localhost, in quanto come anticipato la comunicazione avverrà all'interno della stessa macchina. Di conseguenza inizializziamo il buffer e la stringa "data" che utilizzeremo in seguito per il parsing dei dati. La seconda sezione di codice riportata qui sotto mostra invece il loop all'interno del quale entriamo per ricevere lo stream dei dati.

```
// Enter the listening loop.
while (true)
{
    Thread.Sleep(10);
    Debug.Log("Waiting for a connection... ");
    client = server.AcceptTcpClient();
```

```

if (client != null)
{
    Debug.Log("Connected!");
}
data = null;

// Get a stream object for reading and writing
stream = client.GetStream();
StreamWriter swriter = new StreamWriter(stream);
int i;

// Loop to receive all the data sent by the client.
while ((i = stream.Read(bytes, 0, bytes.Length)) != 0)
{
    Thread.Sleep(50);
    try
    {
        data = System.Text.Encoding.ASCII.GetString(bytes, 0, i);
    }
}

```

Poiché la gestione dei movimenti deve avvenire in concorrenza alla ricezione dei dati, si utilizzano Thread per la gestione del socket come mostrato nel codice, in questo modo sfruttiamo l'hardware a disposizione per garantire la parallelizzazione delle funzioni.

### 7.1.2 Parsing del dato

In parallelo alla ricezione dello stream di bytes, utilizziamo la stringa "data" per modificare il formato delle informazioni ottenute, il valore ricevuto deve quindi essere parsato e controllato in quanto può capitare di ricevere valori non consistenti per ragioni non chiare.

```

if (!data.Contains(" "))
{
    Debug.Log("pacchetto errato");
}
else
{
    var str = data.Substring(data.IndexOf(' ') + 1);
    float val = float.Parse(str);
    if (val >= 0.0 && val <= 1.0)
    {
        value = val;
        value2 = (1 - value);
    }
}

```

}

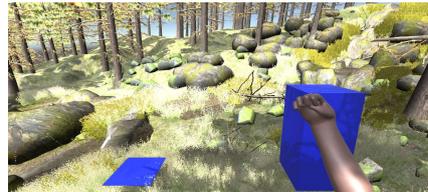
In particolare, i pacchetti errati sono tutti caratterizzati dalla presenza di spazi, possiamo quindi facilmente introdurre un controllo per la verifica della validità del dato ricevuto. Inoltre, il valore è sempre compreso tra 0.0 ed 1.0, legati relativamente alla classificazione di mano destra e mano sinistra, se il dato supera il valore 1.0 o scende sotto il valore 0.0, allora il pacchetto non viene preso in considerazione.

### 7.1.3 Animazioni e Animator Controller

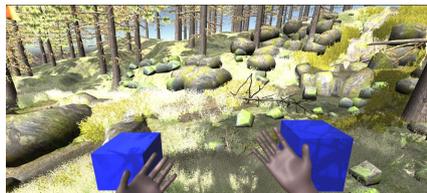
Dopo aver gestito la ricezione dei dati possiamo procedere attraverso l'interfaccia grafica di unity nello strutturare la scena nella quale il partecipante sarà protagonista. In particolare gli arti sono modelli 3d dotati di cinematica, è quindi possibile modificare singolarmente la posizione di ogni articolazione per rendere fluide le animazioni. L'utilizzo dell'Animator Controller presente in unity ci permette di attivare le animazioni in particolari condizioni, nel mio caso ho deciso d'impostare una soglia di cut-off così da attivare le animazioni nel momento in cui il classificatore discrimina l'intenzione del partecipante.



*Classificazione della mano sinistra.*



*Classificazione della mano destra.*



*Durante le transizioni, le mani sono aperte*

Infine, come visibile, ho deciso di posizionare il partecipante al centro di una scena realistica arricchita di suoni per massimizzare l'immersione dell'utilizzatore.

## Capitolo 8

### Conclusioni e prospettive future

Gli avanzamenti più recenti nel campo delle BCI ha permesso l'applicazione di questa tecnologia in diversi contesti, e risulta sempre maggiore l'interesse da parte di diverse compagnie per questa innovativa tecnologia. Tuttavia, benché la ricerca stia compiendo grandi passi in questo emergente campo di studio, al giorno d'oggi non siamo ancora in grado di sviluppare sistemi perfettamente funzionanti e mobili che ne permettano l'utilizzo totalmente incondizionato da parte delle persone, anche se compagnie come Neuralink cercano attraverso i loro strumenti di riuscire in questo scopo. L'obiettivo di questo studio è stato quello d'implementare uno dei principali approcci discriminatori legati alla Motor Imagery, osservando le principali problematiche che ne possono derivare e i possibili punti di forza che la caratterizzano. Lo studio ha dimostrato come sia possibile, attraverso l'utilizzo di piattaforme hardware e software, la discriminazione asincrona dell'attività neuronale in condizioni real-time, raggiungendo valori di accuratezza del 80%.



Figura 8.1: Test dell'applicazione attraverso la strumentazione messa a disposizione dal Centro di ricerca Enrico Piaggio di Pisa.

Tuttavia, è stato osservato come l'applicazione dell'headset in maniera non conforme, crea delle importanti variazioni in termini di accuratezza, comportandone un'impossibilità di applicazione soprattutto nel caso di feedback visivo attraverso la realtà virtuale, in cui i movimenti del partecipante creano artefatti, in particolare di origine muscolare, all'interno del tracciato EEG riducendo drasticamente le performance del classificatore. Recenti innovazioni hanno portato alla nascita di una strumentazione più adeguata, come ad esempio l'headset Galea prodotto dall'azienda OpenBCI che integra un visore di realtà virtuale per lo sviluppo di studi come quello presentato all'interno di questa tesi. I futuri Sviluppi di questo sistema prevedono l'estensione dell'algoritmo implementato al caso di più classi, cercando di aumentare il numero di segnali di controllo al fine di conformare applicazioni più complesse.

## Bibliografia

- [1] Sarah N. Abdulkader, Ayman Atia, and Mostafa-Sami M. Mostafa. Brain computer interfacing: Applications and challenges. *Egyptian Informatics Journal*, 16(2):213–230, 2015.
- [2] Faith A. Bazley, Charles Hu, Anil Maybhate, Amir Pourmorteza, Nikta Pashai, Nitish V. Thakor, Candace L. Kerr, and Angelo H. All. Electrophysiological evaluation of sensory and motor pathways after incomplete unilateral spinal cord contusion. *Journal of neurosurgery. Spine*, 16:414–23, Apr 2012.
- [3] Francesco Ferracuti, Valentina Casadei, Ilaria Marcantoni, Sabrina Iarlori, Laura Burattini, Andrea Moneriù, and Camillo Porcaro. A functional source separation algorithm to enhance error-related potentials monitoring in noninvasive brain-computer interface. *Computer Methods and Programs in Biomedicine*, 191:105419, 2020.
- [4] G. Pfurtscheller. Functional brain imaging based on erd/ers. *Vision Research*, 41(10):1257–1260, 2001.
- [5] Yongwoong Jeon, Chang S. Nam, Young-Joo Kim, and Min Cheol Whang. Event-related (de)synchronization (erd/ers) during motor imagery tasks: Implications for brain–computer interfaces. *International Journal of Industrial Ergonomics*, 41:428–436, 2011.
- [6] Herbert H. Jasper. Report of the committee on methods of clinical examination in electroencephalography. *Electroencephalography and Clinical Neurophysiology*, (10):370–375, 1958.
- [7] Valer Jurcak, Daisuke Tsuzuki, and Ippeita Dan. 10/20, 10/10, and 10/5 systems revisited: Their validity as relative head-surface-based positioning systems. *NeuroImage*, 34(4):1600–1611, 2007.
- [8] E.C. Leuthardt, G. Schalk, J.R. Wolpaw, J.G. Ojemann, and D.W. Moran. A brain-computer interface using electrocorticographic signals in humans. *Journal of Neural Engineering*, 1(2):63–71, 2004. cited By 827.
- [9] G.K. Aguirre, J.A. Detre, E. Zarahn, and D.C. Alsop. Experimental design and the relative sensitivity of bold and perfusion fmri. *NeuroImage*, 15(3):488–500, 2002.

## Bibliografia

- [10] Jaeyoung Shin, Jinuk Kwon, and Chang-Hwan Im. A ternary hybrid eeg-nirs brain-computer interface for the classification of brain activation patterns during mental arithmetic, motor imagery, and idle state. *Frontiers in Neuroinformatics*, 12:5, 2018.
- [11] Hongtao Wang, Yuanqing Li, Jinyi Long, Tianyou Yu, and Zhenghui Gu. An asynchronous wheelchair control by hybrid eeg-eog brain-computer interface. *Cognitive neurodynamics*, 8:399–409, Oct 2014.
- [12] Vince D. Calhoun and Tülay Adalı. Multisubject independent component analysis of fmri: a decade of intrinsic networks, default mode, and neurodiagnostic discovery. *IEEE reviews in biomedical engineering*, 5:60–73, 2012.
- [13] Benjamin Wittevröngel and Marc M. Van Hulle. Spatiotemporal beamforming: A transparent and unified decoding approach to synchronous visual brain-computer interfacing. *Frontiers in Neuroscience*, 11:630, 2017.
- [14] M. Jeannerod. The representing brain: Neural correlates of motor intention and imagery. *Behavioral and Brain Sciences*, 17(2):187–202, 1994.
- [15] Blackwood D. H. and Muir W. J. Cognitive brain potentials and their application. *The British Journal of Psychiatry*, (157):96–101, 1990.
- [16] Aya Rezeika, Mihaly Benda, Piotr Stawicki, Felix Gembler, Abdul Saboor, and Ivan Volosyak. Brain-computer interface spellers: A review. *Brain sciences*, 8, Mar 2018.
- [17] Korbinian Brodmann. *Vergleichende Lokalisationslehre der Grosshirnrinde in ihren Prinzipien dargestellt auf Grund des Zellenbaues*. Barth, 1909.
- [18] G. di Pellegrino, L. Fadiga, L. Fogassi, V. Gallese, and G. Rizzolatti. Understanding motor events: a neurophysiological study. *Experimental Brain Research*, 91(1):176–180, 1992.
- [19] L F Haas. Hans berger (1873–1941), richard caton (1842–1926), and electroencephalography. *Journal of Neurology, Neurosurgery & Psychiatry*, 74(1):9–9, 2003.
- [20] SCCN. Lab Streaming Layer. <https://github.com/sccn/labstreaminglayer>, 2008.
- [21] Delorme A Makeig S. Eeglab: an open-source toolbox for analysis of single-trial eeg dynamics. *Journal of Neuroscience Methods*, pages 9–21, 2004.
- [22] T-P. Jung S. Makeig, A.J. Bell and T.J. Sejnowski. Independent component analysis of electroencephalographic data. *Advances in Neural Information Processing Systems*, 1996.

- [23] Berkhout J. O'Donnell, R. D. and W. R. Adey. Contamination of scalp eeg spectrum during contraction of cranio-facial muscles. *Electroencephalogr. Clin. Neurophysiol.*, 1974.
- [24] Aysa Jafarifarmand and Mohammad Ali Badamchizadeh. Real-time cardiac artifact removal from eeg using a hybrid approach. In *2018 International Conference BIOMDLORE*, pages 1–5, 2018.
- [25] Steven Lemm Motoaki Kawanabe Benjamin Blankertz, Ryota Tomioka and Klaus-Robert Mueller. Optimizing spatial filters for robust eeg single-trial analysis. *IEEE Signal Process Mag*, pages 41–56, 2008.
- [26] R. Fisher. The use of multiple measurements in taxonomic problems. *Annals Eugen*, pages 179–188, 1936.
- [27] Fatma Zohra Chelali, A. Djeradi, and R. Djeradi. Linear discriminant analysis for face recognition. In *2009 International Conference on Multimedia Computing and Systems*, pages 1–10, 2009.
- [28] Chunjun Zheng and Yi Zhang. Image retrieval based on lda and svm. In *2019 11th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, pages 446–449, 2019.
- [29] Christian Kothe and Scott Makeig. Bcilab: A platform for brain-computer interface development. *Journal of neural engineering*, 10:056014, 08 2013.
- [30] Pietro Cipresso, Irene Alice Chicchi Giglioli, Mariano Alcañiz Raya, and Giuseppe Riva. The past, present, and future of virtual and augmented reality research: A network and cluster analysis of the literature. *Frontiers in psychology*, 9:2086, 2018.