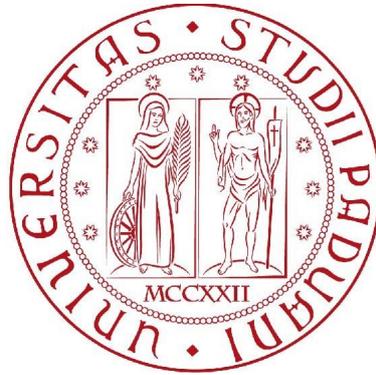


Università degli studi di Padova
Facoltà d'Ingegneria
Magistrale di Bioingegneria



Uso di tecniche bayesiane per il miglioramento
delle performance in un sistema BCI basato
sulla P300

Relatore

Prof. Giovanni Sparacino

Correlatori

Ing. Costanza D'Avanzo

Ing. Anahita Goljahani

Ing. Stefano Silvoni

Laureanda: Clara Genna

Anno Accademico 2010/2011

alla mia mamma, amica di sempre

Sommario

Il sistema BCI (Brain-Computer Interface) è un'interfaccia in grado di realizzare la comunicazione, per esempio di soggetti con importanti patologie muscolari e neurologiche, e la possibilità di pilotare dispositivi esterni, per esempio di soggetti con arti mancanti, senza la mediazione dei muscoli e dei nervi ma attraverso l'analisi diretta dell'attività celebrale.

Il segnale dell'attività celebrale largamente utilizzato in ambito BCI è il tracciato EEG. Diverse componenti d'interesse possono essere impiegate, come gli slow cortical potentials, la P300 ed altri potenziali evento-correlati, i ritmi μ ed i ritmi beta. Tutti questi segnali, secondo proprie caratteristiche fisiologiche, sono in grado di trasportare l'informazione relativa alla volontà del soggetto.

La componente d'interesse in questo lavoro è la P300, un potenziale ERP che viene elicitato quando lo stimolo presentato ha una certa rilevanza per il soggetto. Si parla così di paradigma oddball, una sequenza di stimoli in cui eventi rari, i quali elicitano la P300, sono alternati ad eventi più frequenti.

Il sistema BCI analizzato in questa tesi è quello presente nell'istituto IRCCS San Camillo Venezia. Esso è un sistema BCI basato sulla P300. Il sistema è in grado di individuare la presenza od assenza della P300 e, in funzione di questo, fornire l'adeguato feedback. In particolare, tale riconoscimento prevede di separare la componente ERP dal rumore rappresentato dal segnale EEG di fondo e successivamente, sulla base di tale stima, applicare la procedura di estrazione delle features e il classificatore.

In questa tesi ci si vuole soffermare proprio sul blocco di elaborazioni ed in particolare sull'introduzione di nuove tecniche di analisi single trial, per valutare se, una diversa elaborazione del segnale, è in grado di migliorare l'identificazione della componente ERP in termini di performance del classificatore.

Le due tecniche che verranno testate sono un metodo bayesiano a due step (B2S) ed un metodo di Multi Task Learning (MTL). Entrambi i metodi si basano sulla stima bayesiana. Il metodo B2S si basa su due passi. Il primo step è il calcolo dell'ERP medio, ovvero il profilo atteso della P300; il secondo step è il calcolo della stima finale vera e propria. Il metodo MTL stima la componente ERP dall'analisi congiunta di tutte le tracce acquisite da uno stesso soggetto e che costituiscono così la popolazione omogenea.

Quanto svolto in questo lavoro è definito *validazione off-line*. A partire da dati acquisiti in precedenti sessioni sperimentali, si eseguiranno delle prove applicando i medesimi step presenti nel sistema BCI senza che quest'ultimo sia in funzione. Si dimostrerà che l'utilizzo di tecniche bayesiane per il pre-processing del segnale è in grado di migliorare le performance del sistema, soprattutto in termini di errore sul riconoscimento della P300.

La tesi è così strutturata. Nel capitolo 1 si introducono i sistemi BCI e segnali di controllo utilizzati, esaminando lo schema a blocchi di un generico sistema BCI basato su EEG e nel dettaglio quelli con segnale di controllo la P300. Si passa così all'analisi del sistema BCI del IRCCS San Camillo (capitolo 2). Dopo l'introduzione teorica del metodo B2S (capitolo 3) e del metodo MTL (capitolo 4), si passa all'analisi delle stime nonché delle performance. Due sono i filoni principali: applicazione delle metodiche ai dati simulati ed ai dati reali. Nel capitolo 5 si esegue prima la

descrizione del processo di simulazione dei dati e successivamente l'analisi delle stime ottenute con il metodo B2S ed il metodo MTL. Nel capitolo 6 si introducono le stime per i dati reali con i due metodi. Nel capitolo 7, si addestreranno i classificatori sulla base delle features estratte da queste nuove stime. Infine si analizzano gli errori di classificazione rispetto ai dati simulati ed ai dati reali. Nel capitolo 8 verranno riportate le conclusioni e quindi si suggeriscono alcuni possibili sviluppi futuri.

Indice

Indice	7
1 Sistemi di Brain-Computer Interface (BCI)	11
1.1 Introduzione ai sistemi di Brain-Computer Interface	11
1.2 Descrizione delle diverse tipologie di BCI	14
1.3 Analisi dello schema a blocchi per un generico sistema BCI basato su EEG	18
1.3.1 Descrizione della procedura di l'acquisizione del segnale EEG	19
1.3.2 Descrizione delle tecniche di pre-processing del segnale	23
1.3.3 Descrizione del processo di estrazione delle features	24
1.3.4 Descrizione della procedura di classificazione	25
1.4 Indici di validazione del sistema BCI	25
1.5 Brain Computer Interface basata su P300	26
1.5.1 Analisi dei processi di generazione dell'onda P300	27
1.5.2 Descrizione dell'Interfaccia Grafica per l'elicitazione della P300 nei sistemi BCI	30
1.6 Conclusioni	32
1.7 Scopo della tesi	32
2 Il sistema Brain Computer Interface dell'IRCCS San Camillo	35
2.1 Il Paradigma Oddball	35
2.2 Schema a blocchi del sistema	36
2.2.1 Blocco di acquisizione del segnale EEG	37
2.2.2 Blocco di pre-elaborazione del segnale	40
2.2.3 Blocco di features extraction	41
2.2.4 Blocco di classificazione	42
2.3 Protocollo sperimentale applicato per l'acquisizione del data base	42
2.4 Conclusioni	44
3 Un metodo bayesiano di stima single-trial a due passi (B2S)	45
3.1 Approccio generale alla stima bayesiana a minima varianza d'errore	45
3.2 Assunzioni del metodo bayesiano a due passi	46
3.2.1 Descrizione dell'errore di misura	47
3.2.2 Descrizione dell'informazione a priori del segnale incognito $u_j(t)$	48
3.2.3 Calcolo del vettore di riferimento μ	49
3.3 Formulazione finale e parametro di regolarizzazione	49

3.4	Conclusioni	50
4	Un metodo di Multi Task Learning per la stima single-trial (MTL)	53
4.1	Introduzione al metodo MTL	53
4.2	Assunzioni del Multi Task Learning	53
4.3	Stima della curva media $\bar{z}(t)$	55
4.4	Stima delle curve individuali $z_j(t)$	56
4.5	Modelli di Wiener e determinazione degli iper-parametri	57
4.6	Conclusioni	57
5	Applicazione delle tecniche B2S e MTL per la stima della P300 da dati simulati	59
5.1	Generazione dei dati simulati	59
5.1.1	Simulazione delle sweeps non target	59
5.1.1.1	Modelli AR per la simulazione del rumore	59
5.1.1.2	Generazione delle sweeps non target	60
5.1.2	Simulazione delle sweeps target	61
5.1.2.1	Costruzione della template	61
5.1.2.2	Generazione delle sweeps target	63
5.1.3	Scelta del pre-stimolo per la simulazione dei dati	64
5.1.3.1	Caso I: prestimolo estratto da dati grezzi	64
5.1.3.2	Caso II: prestimolo da sweeps filtrate a 15 Hz	66
5.1.3.3	Considerazioni finali	67
5.2	Metodo B2S: risultati	68
5.2.1	Casi particolari	72
5.3	Metodo MTL: risultati	74
5.3.1	Casi particolari	77
5.4	Conclusioni	78
6	Applicazione delle tecniche B2S e MTL per la stima della P300 da dati reali	81
6.1	Costruzione del dataset	81
6.2	Metodo B2S: risultati	82
6.2.1	Casi particolari	86
6.3	Metodo MTL: risultati	87
6.3.1	Casi particolari	91
6.4	Conclusioni	93
7	Valutazione off-line del sistema BCI con tecniche di pre-processing bayesiane	95
7.1	Schema di addestramento del classificatore	95
7.2	Valutazione delle performance del sistema BCI	96
7.3	Risultati dai dati simulati	98
7.4	Risultati dai dati reali	99
7.5	Conclusioni	105

8	Conclusioni e sviluppi futuri	107
8.1	Bilancio e considerazioni riassuntive	107
8.2	Sviluppi futuri	108
8.2.1	Miglioramento della stima single trial con il metodo B2S	108
8.2.2	Analisi della stime da sweeps non target con i metodi B2S ed MTL	111
8.2.3	Analisi delle performance con un nuovo schema di addestramento .	115
A	Analisi delle stime MTL con filtraggio a 15 Hz dai dati Simulati	119
B	Codice Matlab	125
	Bibliografia	139

Capitolo 1

Sistemi di Brain-Computer Interface (BCI)

1.1 Introduzione ai sistemi di Brain-Computer Interface

Una BCI [3] è un sistema in grado di impostare una linea diretta fra il cervello del soggetto ed il computer, fornendo un nuovo canale di controllo e comunicazione. Con un sistema di questo tipo, i soggetti con importanti patologie neuromuscolari, i quali non hanno possibilità di comunicare con i normali canali sensoriali, sono in grado di far convergere i loro messaggi al mondo esterno. La figura 1.1.1 mostra un generico sistema BCI.

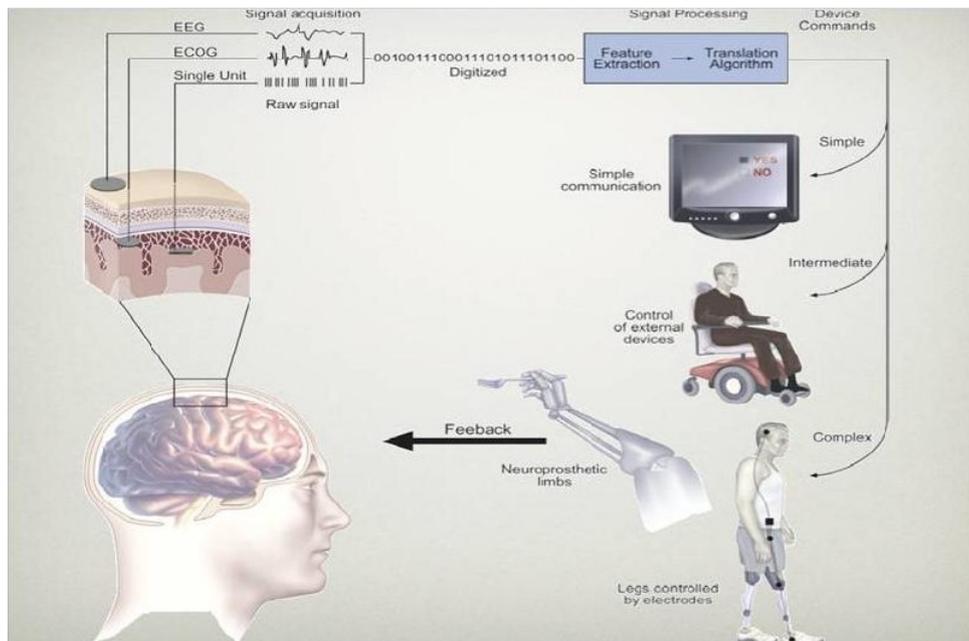


Figura 1.1.1: Esempio di un generico sistema BCI [s4].

La volontà del soggetto di compiere una determinata azione è adesso contenuto “in codice” nel segnale di attività celebrale che viene acquisito, quindi digitalizzato ed infine elaborato dallo stesso sistema BCI (blocco *Signal Processing* di figura 1.1.1), fornendo un output. Questo è poi tradotto in feedback fornito al soggetto, secondo modalità dipendenti dallo specifico sistema implementato. Tale feedback ha un ruolo fondamentale nel sistema BCI. Infatti, i normali canali neuromuscolari di output del cervello eseguono correttamente delle operazioni grazie al processo congenito di feedback. Azioni come parlare, camminare o ballare vengono eseguite correttamente dal nostro corpo proprio perché il movimento determina modificazioni nelle afferenze sensoriali provenienti dai muscoli, tendini, articolazioni e dalla cute, le quali rappresentano le informazioni che rendono esatto e fluido il movimento [15]. Quindi, anche il sistema BCI è dipendente dal feedback e dall’adattamento dell’attività celebrale al feedback fornito. Questo significa che le operazioni BCI dipendono da due controllori adattativi: da un lato il cervello del soggetto, il quale produce i segnali misurati dalla BCI; dall’altro la stessa BCI che trasla questi segnali in specifici comandi [43]. Nel primo controllore, è di fondamentale importanza la modalità con cui il sistema BCI è in grado di far generare al cervello i segnali, ovvero il *task*. Un tipico esempio sono i paradigmi in grado di elicitare le componenti d’interesse. In quest’ambito si utilizzano specifiche interfacce grafiche ed il soggetto dovrà porre la sua attenzione verso un preciso stimolo (target), ignorando così i rimanenti stimoli non target (per esempio il *paradigma oddball*). Un’altra tipologia di stimolazione è la *motor imagery* [27]. Il soggetto pensa di voler eseguire un movimento, come prendere un libro su un tavolo, senza che questo sia generato fisicamente come output. Studi condotti hanno dimostrato che la rappresentazione mentale di un movimento coinvolge le stesse risorse cerebrali di quando il movimento viene realmente effettuato. Per quanto concerne il feedback vero e proprio, questo dipende, come detto, dallo specifico sistema BCI implementato. In riferimento alla figura 1.1.1, il livello indicato con *simple* si riferisce ad esempio al muovere un cursore sullo schermo; l’*intermediate* è la condizione in cui l’output viene utilizzato per interpretare gli spostamenti delle carrozzine in ambienti reali; lo stato *complex* si riferisce invece alla possibilità che il soggetto ha di pilotare esso stesso le sue protesi. Inoltre, la ricerca sta esplorando anche un altro ambito: BCI per riabilitazione [21]. L’approccio è il medesimo del livello *complex*, con la differenza che il soggetto non comanda la protesi ma “viene comandato” dallo strumento riabilitativo, come ad esempio un braccio meccanico che invia un feedback di forza.

Si intuisce così che gli ambiti di applicazione sono diversi e diversificati nelle strategie implementative. L’uso più comune del sistema BCI, nonché quello trattato in questa tesi, è il livello *simple* della figura 1.1.1, ovvero la semplice comunicazione. Quest’ambito è fortemente dedicato a tutti quei soggetti con severe malattie muscolari degenerative, come la ALS (sclerosi laterale amiotrofica), gli ictus cerebrali o le lesioni spinali.

I dati utilizzati per le elaborazioni di questa tesi sono stati acquisiti durante protocolli sperimentali eseguiti da soggetti ALS. L’ALS [22] è una malattia degenerativa e progressiva del sistema nervoso che colpisce selettivamente i cosiddetti neuroni di moto (motoneuroni), sia centrali - 1^o motoneurone a livello della corteccia cerebrale, sia periferici - 2^o motoneurone, a livello del tronco encefalico e del midollo spinale. In media, dopo 3-5 anni dalla diagnosi della malattia, il paziente ALS inizia a perdere il controllo di tutti i muscoli

volontari, ad eccezione degli occhi, divenendo tetraplegico e anartrico, ovvero privo della parola, mantenendo però intatte le proprie capacità intellettive, cognitive ed emozionali. Questo stadio, di livello intermedio, [22] è detto *locked-in state* (LIS); mentre, l'ultimo stadio è il *completed locked-in* (CLIS), in cui il paziente perde anche il movimento degli occhi e la respirazione autonoma.

Per dare un'idea della diffusione nel mondo di questa patologia, in figura 1.1.2 è riportato un atlante del 2008 realizzato dalla *World Health Organization* e *multiple sclerosis international federation* [s5]. In particolare, è mostrata la diffusione della malattia in relazione all'età. E' da notare come la fascia di età maggiormente diffusa è quella compresa fra i 20 e 35 anni. Sono soggetti molto giovani, i quali dovranno convivere a lungo con questa condizione. Lo studio di sistemi BCI diventa così un punto cardine in quest'ottica.

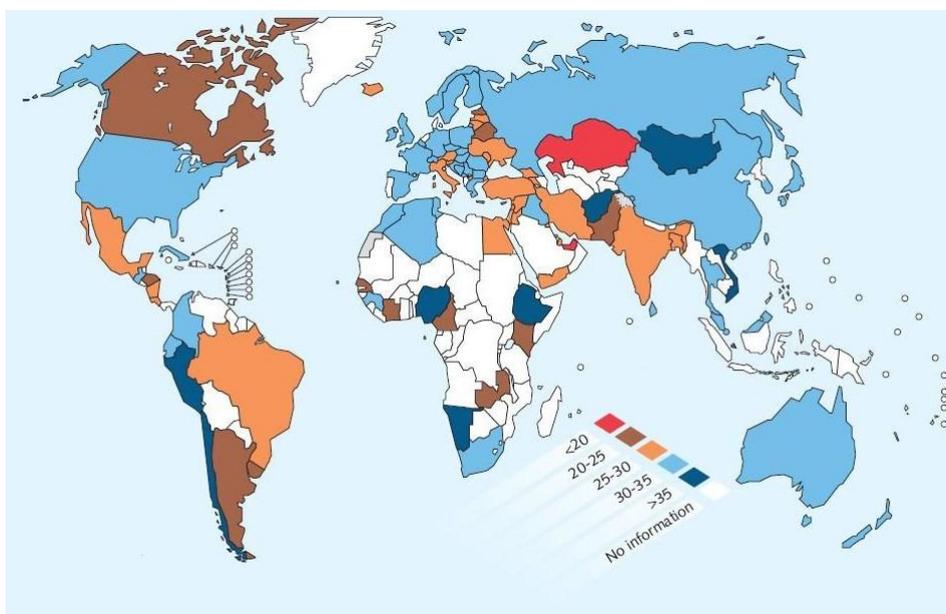


Figura 1.1.2: Atlante ALS nel mondo [s5].

Un'interessante implementazione di un sistema BCI per pazienti ALS è descritto in [25], in cui tutta la fase di sperimentazione viene svolta non in ospedale o nei laboratori, bensì direttamente a casa del paziente. Questo è la direzione verso cui si muove la ricerca BCI: fornire un *tool* di comunicazione indipendente dal controllo dei muscoli significa dotare i pazienti LIS di un livello di autonomia non indifferente, rendendoli parzialmente indipendenti nella comunicazione, soprattutto quando l'unico movimento ad essi permesso è quello degli occhi, ma soprattutto realizzare un sistema che sia effettivamente utilizzabile quotidianamente dal soggetto comodamente nella propria casa.

1.2 Descrizione delle diverse tipologie di BCI

L'input del sistema BCI è un generico segnale di attività celebrale, il quale, dopo esser stato digitalizzato, viene elaborato secondo opportune tecniche di pre-processing.

Proprio sulla base del DOVE e COME il segnale di attività celebrale viene prelevato, è possibile distinguere diverse tipologie di Brain-Computer Interface.

La principale suddivisione dei sistemi BCI descritta in [3], è determinata sulla base del DOVE il segnale dell'attività celebrale viene prelevato. Secondo questo criterio si distinguono *BCI invasiva* e *BCI non invasiva*.

Si parla di *BCI invasiva* [3] quando il segnale viene prelevato da micro o macro elettrodi impiantati direttamente nel cervello.

I segnali di controllo più diffusi sono i seguenti

1. Potenziali sinaptici ed extra-cellulari [44, 3]. Sono segnali che hanno un'elevata risoluzione sia temporale che spaziale. Fanno riferimento ai singoli potenziali d'azione neuronali, detti *spikes*, oppure LFPs, i quali riflettono l'attività combinata fra neuroni vicini e relative sinapsi.
2. Elettrocorticogramma (ECoG) [3, 44]. L'elettrocorticografia [1] è stata sperimentata per la prima volta all'inizio degli anni '50 dai neurochirurghi canadesi Wilder Penfield e Herbert Jasper. I potenziali corticali registrati con l'ECoG furono usati per identificare le zone epilettogene, ossia le aree della corteccia cerebrale responsabili dell'origine delle crisi epilettica. Il segnale ECoG è generato dai potenziali [24] che si presentano in principio sui neuroni corticali piramidali e poi vengono diffusi attraverso vari strati e membrane, in modo perpendicolare alla corteccia cerebrale, prima di raggiungere gli elettrodi subdurali utilizzati nella registrazione, posizionati appena sotto la dura madre, membrana del cranio più esterna. Per impiantare gli elettrodi è dunque necessaria un'operazione chirurgica di craniotomia che rimuova parte del cranio e apra una finestra sulla stessa, come indicato in figura 1.2.1.

Per questa tipologia di BCI, i principali lavori presentati fanno riferimento a cavie animali per l'evidente invasività della tecnica e la scarsa ripetibilità dell'esperimento. Tra l'altro nasce anche un problema di durata degli elettrodi impiantati, oltre al fatto che l'uso continuo per lunghi periodi può provocare il danneggiamento della parte esposta [44].

Invece, si parla di *BCI non invasiva* [3] quando il segnale viene prelevato da elettrodi posti sullo scalpo. Il segnale EEG standard [24] è generato dai medesimi potenziali del segnale ECoG. Ma, a differenza di questi, tali potenziali elettrici devono essere condotti attraverso l'osso del cranio per raggiungere gli elettrodi di superficie posti sullo scalpo, dove vengono notevolmente attenuati a causa della scarsa conduttività dello stesso.

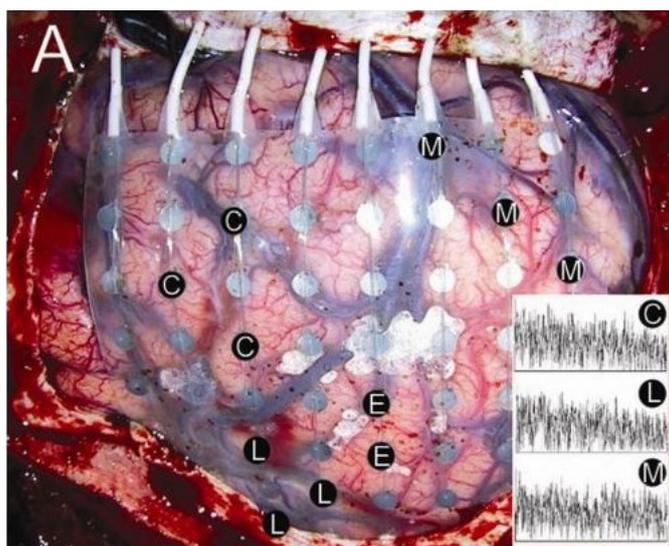


Figura 1.2.1: Esempio di acquisizione del segnale ECoG [1].

I segnali largamente utilizzati sono

1. *Slow Cortical Potential (SCP)*

I potenziali corticali lenti, *Slow Cortical Potential (SCP)* [1, 43, 24], sono lente variazioni di potenziale registrate a livello della corteccia cerebrale, che presentano una frequenza variabile tra 1 e 2 Hz. Variazioni negative del potenziale corticale vengono associate ad attività motorie e altre funzioni che coinvolgono l'attività corticale, mentre variazioni positive corrispondono ad una diminuzione dell'attività corticale. La loro occorrenza può variare all'interno della finestra temporale 0.5 e 10 secondi. La gestione di questi segnali per sistemi BCI si basa sulla possibilità di addestrate i soggetti nell'arco di pochi mesi a modulare l'intensità del potenziale in funzione della tipologia dello stimolo (figura 1.2.2). In questo caso si parla di *thought translation device (TTD)* [2].

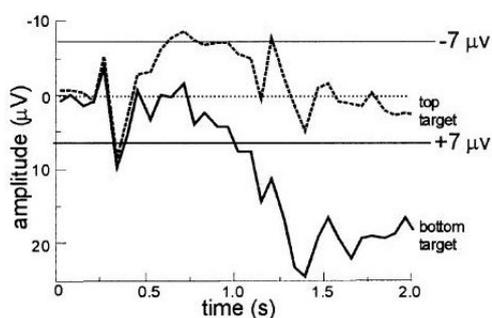


Figura 1.2.2: *Slow Cortical Potentials* [43]. Il segnale tratteggiato è la risposta generata in corrispondenza di un evento target (da raggiungere) posto in alto nella schermo. In linea continua, il segnale generato in corrispondenza di un diverso target posto in basso.

1. SISTEMI DI BRAIN-COMPUTER INTERFACE (BCI)

2. Potenziale evento correlato e P300 (potenziale ERP)

Questa tipologia di segnale verrà descritta in dettaglio successivamente.

3. I ritmi sensorimotori mu e beta

I ritmi sensorimotori [1, 43, 24] sono associati alle aree corticali direttamente legate ai canali motori naturali del cervello. Questi ritmi includono una serie di oscillazioni comprese fra gli 8 Hz e i 32 Hz, ma quelli indagati in ambito BCI sono il ritmo mu fra gli 8-12 Hz e il ritmo beta fra i 18-26 Hz. Quello che accade è che in corrispondenza del movimento o della semplice preparazione ad esso si ha un decremento della positività dei ritmi mu e beta, fenomeno chiamato *Event-Related Desynchronization* (ERD). Successivamente, in corrispondenza del rilassamento si registra un aumento della potenza dei suddetti ritmi, chiamato invece *Event-Related Synchronization* (ERS). La peculiarità di questi segnali è che essi sono naturalmente legati alle aree cerebrali del movimento. Questo rende applicabili i ritmi mu e beta come segnali di controllo per BCI non solo con scopo di comunicazione, ma anche riabilitativo e come segnale utile per pilotare protesi, in quanto essi vengono generati non solo in corrispondenza dell'azione eseguita, ma anche dalla sola immaginazione di essa. Anche in questo caso, nell'applicazione del sistema BCI, un opportuno addestramento del soggetto permette di potenziare questi segnali. Si fornisce un esempio del ritmo mu in figura 1.2.3. Si nota come l'ampiezza in frequenza sia prominente verso l'evento target posto in alto nello schermo (*top target*) e minima in riferimento all'evento target posto in basso (*bottom target*).

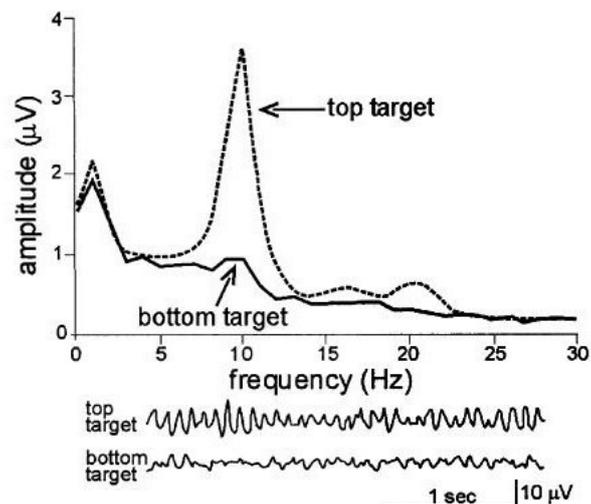


Figura 1.2.3: *Ritmo Mu* [43]. L'immagine indica la diversa ampiezza del ritmo mu in funzione dello stimolo presentatosi al soggetto.

Altre tecniche di BCI non invasive [3, 44] sono la tomografia ad emissione di positroni (PET), la risonanza magnetica funzionale (fMRI), con particolare attenzione alla risposta BOLD, la misura del flusso del sangue corticale nel vicino infrarosso (NIRS) ed infine la

magnetoencefalografia (MEG). Il sistema BCI basato su fMRI, PET, NIRS [44] misura i cambiamenti nella risposta emodinamica del cervello. Queste tecniche si caratterizzano per l'elevata risoluzione spaziale, ma sono carenti in termini di risoluzione temporale. Inoltre, sono dispositivi molto costosi e tecnicamente complessi.

La MEG [44] misura invece l'attività magnetica del cervello, permettendo un controllo in tempo reale oltre che un'eccellente risoluzione temporale e spaziale. Anche questa tecnica però è molto costosa ed impraticabile per continui esperimenti clinici.

La più diffusa tipologia di BCI sono, tra le invasive, quella basata su ECoG e tra le non invasive quella basata sull'EEG. Anche se la risoluzione spaziale del ECoG è molto più alta di quella del segnale EEG, ECoG è una tecnica notevolmente invasiva e la sua applicazione per esperimenti ripetuti è fortemente limitata. Tanto meno pensabile è l'implementazione di una tecnica con segnale elettrocorticografico per un dispositivo commerciale.

Il segnale elettroencefalografico (EEG) è quindi l'alternativa largamente utilizzata proprio per la sua non invasività e versatilità nelle applicazioni. Per tale motivo, nel successivo paragrafo si farà riferimento a questa tipologia di BCI (BCI basata su EEG).

Un'altra suddivisione per le tipologie di BCI, meno menzionata in letteratura, è inerente a COME il segnale dell'attività celebrale viene "prodotto" [43]. Si distinguono perciò *BCI indipendente* e *BCI dipendente*.

Si parla di *BCI indipendente* [43] quando è il cervello a generare autonomamente il segnale che trasporta il messaggio da interpretare, come nel caso della componente P300 (il relativo sistema BCI verrà descritto nella sezione dedicata).

Invece, si parla di *BCI dipendente* [43] quando non si utilizzano le normali vie di uscita dell'encefalo per trasportare il messaggio (o comando), ma c'è bisogno dell'attività di queste vie per generare i segnali cerebrali necessari al controllo, come il caso dell'SSVEP (*Steady State Visual Evoked Potential*).

- Il potenziale evocato visivo [1, 43, 24] è un potenziale che riflette i meccanismi elettrofisiologici del processo di elaborazione dell'informazione visiva a livello cerebrale. A seconda della frequenza di stimolazione, la risposta evocata può essere transiente, frequenza < 2 Hz, o di tipo steady-state (frequenza > 6 Hz). Nel contesto BCI è utilizzata il potenziale *Steady State Visual Evoked Potential* (SSVEP): al soggetto vengono presentate più sorgenti luminose lampeggianti a frequenze diverse, associando a ciascuna di esse una funzione. Il soggetto è in grado di attivare una funzione fissando la sorgente desiderata. La *VEP-based communication* dipende quindi dall'abilità del soggetto di controllare e mantenere la direzione fissa verso la sorgente scelta. Nella figura 1.2.4 è rappresentato un esempio del potenziale VEP che si genera, con le relative componenti principali.

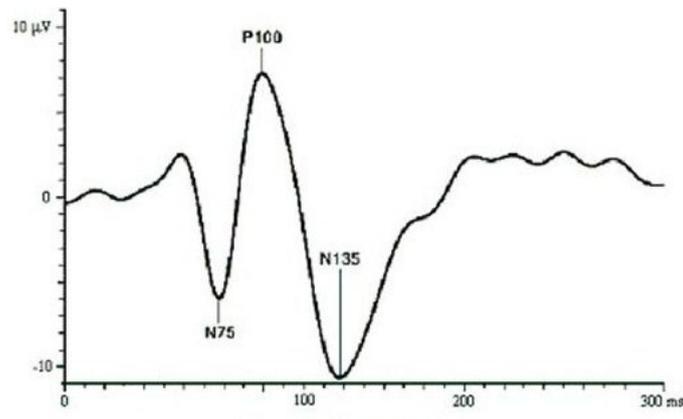


Figura 1.2.4: *Esempio di un Visual Evoked Potentials [s6].*

1.3 Analisi dello schema a blocchi per un generico sistema BCI basato su EEG

Indipendentemente dal protocollo di acquisizione del segnale EEG, lo schema a blocchi di un generico sistema BCI è il medesimo ed è riportato in figura 1.3.1. In questa sezione ci si sofferma sulla descrizione di ogni singolo blocco.

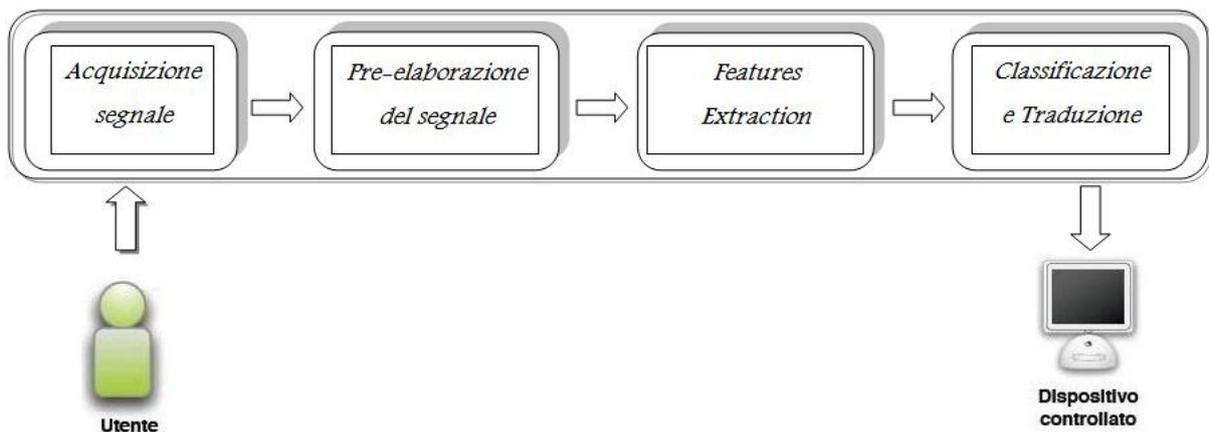


Figura 1.3.1: *Generico schema a blocchi per BCI.*

1.3.1 Descrizione della procedura di l'acquisizione del segnale EEG

Un sistema di acquisizione dati è un sistema di misura elettronico realizzato per monitorare, registrare ed eventualmente elaborare le misure di una o più grandezze fisiche. Come spiegato in precedenza, l'elettroencefalogramma (EEG) [24] è la misura del flusso di corrente extracellulare che viene generato dalla somma delle attività di un elevato numero di neuroni corticali piramidali disposti in corrispondenza dell'area corticale sottostante l'elettrodo.



Nella foto accanto è ritratto Hans Berger, psichiatra tedesco .A lui è dovuta la scoperta, nel 1929, delle registrazioni di deboli correnti elettriche generate nel cervello e rappresentate graficamente su fogli di carta: questa nuova forma di registrazione viene definita elettroencefalogramma (EEG).

La prima registrazione EEG di Hans Berger (circa 1928)

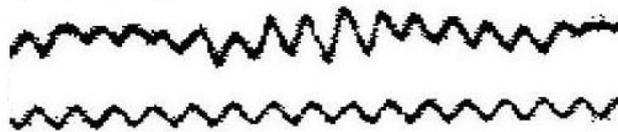


Figura 1.3.2: *Hans Berger* [24].

Il segnale viene acquisito da elettrodi posti sullo scalpo, la cui collocazione è stabilita dal Sistema Internazionale 10-20 [20], indicato in figura 1.3.3.

Ciascuno elettrodo è definito rispetto a due coordinate portanti [15]:

- *area celebrale sottostante*, in cui le diverse zone sono frontale (F), parietale (P), centrale per il vertice (C), temporale (T) ed occipitale (O);
- *linea mediana*, indicata con il numero pari per elettrodi di destra, dispari per elettrodi di sinistra, lettera *z* per quelli collocati sulla linea mediana.

Ad esempio, F3 indica l'elettrodo frontale sinistro; Cz è l'elettrodo centrale sulla linea mediana.

Esistono due tipologie di registrazione: *monopolare* e *bipolare* [15]. Nella monopolare, l'attività di ogni elettrodo può essere riferita ad un comune elettrodo passivo oppure ad una coppia di elettrodi posizionati nel lobo dell'orecchio. La misura sarà la differenza

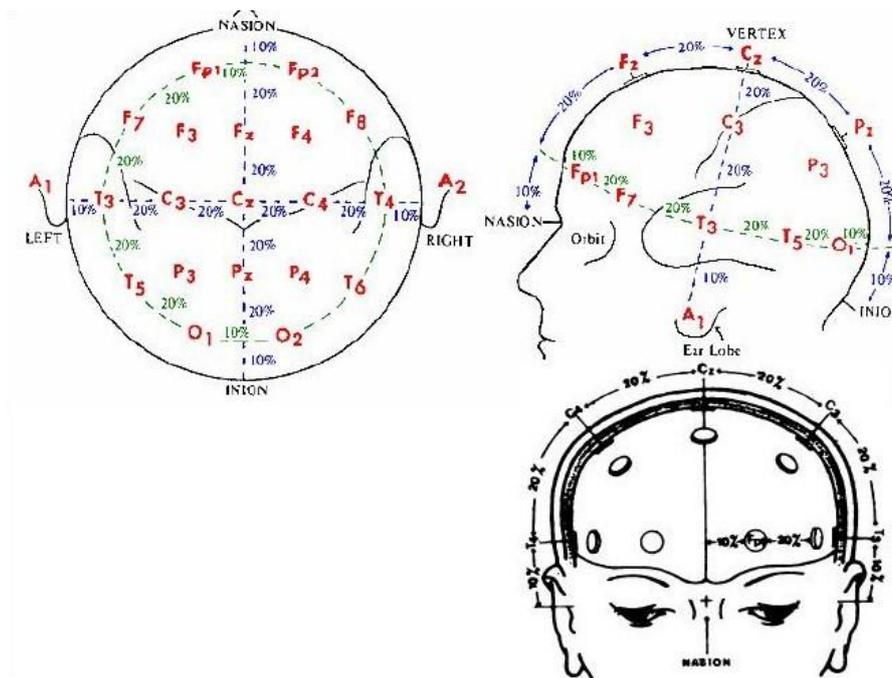


Figura 1.3.3: *Sistema Internazionale 10-20* [1].

di potenziale tra il sito cerebrale sottostante e l'elettrodo di riferimento. Nella bipolare, invece, si registra la differenza di potenziale fra continue coppie di elettrodi.

I protocolli di acquisizione saranno caratterizzati così dal numero e dalle aree in cui gli elettrodi verranno posizionati. Queste scelte sono strettamente dipendenti dal protocollo sperimentale che si decide di seguire, soprattutto inerente al tipo di segnale che si vuole evidenziare.

Per quanto concerne il tipo di elettrodo impiegato, diverse sono le soluzioni. Alcuni esempi sono gli elettrodi a coppetta (in stagno e in argento rivestito di cloruro d'argento (Ag/AgCl), con diametro variabile), gli elettrodi a disco (in argento rivestito di cloruro d'argento (Ag/AgCl) sinterizzati, con diametro variabile), gli elettrodi biadesivi monouso, gli elettrodi ad anello in acciaio inossidabile e gli elettrodi waterproof. Per registrare da un numero considerevole di punti dello scalpo si ricorre all'utilizzo di cuffie realizzate in tessuto elasticizzato (per garantire sempre una perfetta aderenza) con gli elettrodi già incorporati e predisposti secondo la disposizione standardizzata dei siti. In figura 1.3.4 è mostrato un esempio delle suddette tecnologie.

Sono molti gli studi condotti al miglioramento degli elettrodi per l'acquisizione del segnale, soprattutto per la standard procedura di preparazione. Infatti, prima di applicare gli elettrodi, è necessario ridurre la naturale tendenza isolante della pelle pulendo e sgrassando accuratamente la regione d'interesse mediante una pasta abrasiva. Questo ha il duplice scopo di creare un contatto ottimale elettrodo-cute, favorendo in tal modo la conduzione del segnale, e di ammortizzare eventuali movimenti, mantenendo un contatto costante tra pelle ed elettrodo. La procedura di preparazione è indolore e non provoca danni al



Figura 1.3.4: *Esempio di elettrodi a coppetta (sinistra) e di una cuffia per acquisizione EEG (destra) [s14].*

cuoio capelluto. In seguito all'applicazione degli elettrodi, si effettua il controllo delle impedenze elettrodiche, le quali vengono visualizzate attraverso un programma dedicato e devono mantenersi al di sotto dei $5\text{ k}\Omega$ [s8]. Gli elettrodi per cui non si verifica tale condizione, vengono ripassati con la pasta abrasiva.

Questa procedura non è complicata ma piuttosto fastidiosa perché alla fine dell'esperimento il soggetto deve essere ripulito. Per tale motivo, si studia la possibilità di creare qualcosa che sia più veloce, facile e confortevole. E questo è proprio quanto indicato in [42]. La possibilità di utilizzare *water based electrodes* eliminando la parte di preparazione con la pasta abrasiva è chiaramente un vantaggio non trascurabile. L'eliminazione della preparazione del paziente è anche un punto centrale nello sviluppo di sistemi BCI portatili. La possibilità di utilizzare tale dispositivo nella propria casa potendo attaccare e staccare gli elettrodi quando più si reputata necessario senza dover ogni volta ripetere la procedura di preparazione, è chiaramente una libertà importante per il soggetto.



Figura 1.3.5: *Esempio di water based electrodes [42].*

La connessione fra tali sensori, i quale registrano l'attività celebrale, ed il computer di sistema, il quale elabora tale informazione per fornire il corrispondente feedback, sono i comuni fili elettrici, quelli ottici, oppure la nuova generazione di trasmissione dati, ovvero la connessione wireless (Wi-Fi, Bluetooth,...). Quest'ultima ha un aspetto economico più

1. SISTEMI DI BRAIN-COMPUTER INTERFACE (BCI)

alto ed inoltre ha la necessità di dispositivi con maggiore potenza. Questo non è un limite perché nuovi strumenti e tecnologie si susseguono con prestazioni sempre più elevate. Amplificatori e convertitori, quest'ultimi impiegati nella digitalizzazione del segnale acquisito, fanno parte di quella tecnologia ormai assodata, a tal punto da non poter più essere considerata un problema aperto [5]. I sistemi di acquisizione dati possono essere suddivisi in sistemi di tipo PC-based (figura 1.3.6) o sistemi portatili (figura 1.3.7). Nel caso dei sistemi PC-based, la sezione di acquisizione si interfaccia con un PC, che svolge funzioni di elaborazione del segnale e di memorizzazione. L'utente può usufruire di un'interfaccia grafica con la quale controllare e regolare i parametri di acquisizione e di elaborazione ed esaminare i risultati. Un software Matlab interessante è EEG-Lab, il quale consente specifiche analisi del segnale EEG.

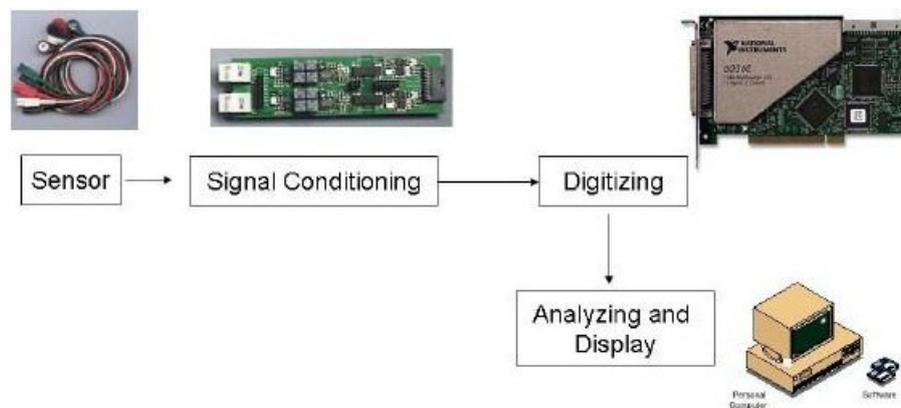


Figura 1.3.6: *Esempio di un sistema PC-based [s10].*

I sistemi di acquisizione dati portatili presentano dimensioni e consumi ridotti e pertanto permettono il monitoraggio dei segnali biologici anche al di fuori dei laboratori clinici. E' così la tecnologia indicata per l'uso di sistemi BCI in ambiente domestico.



Figura 1.3.7: *Esempio di un sistema portatile per acquisizione EEG [s10].*

Il processo di digitalizzazione di un segnale è denominato conversione analogico-digitale (A/D). Il processo inverso tramite il quale da un segnale digitale si ottiene nuovamente un segnale analogico è chiamato conversione D/A. L'elaborazione digitale rispetto a quella analogica presenta caratteristiche di maggiore versatilità e flessibilità, nonché un minor costo. Inoltre la conversione A/D presenta dei vantaggi non solo in termini di elaborazione, ma anche per la memorizzazione e la trasmissione delle informazioni. Il processo di conversione A/D consta di tre stadi. Questi sono il campionamento, la quantizzazione e la codifica numerica, come indicato in figura 1.3.8.

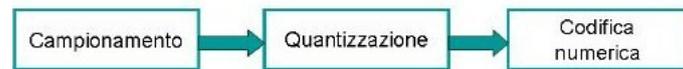


Figura 1.3.8: *Schema a blocchi del processo di conversione A/D.*

Il campionamento è l'operazione che consente di discretizzare l'asse temporale del segnale analogico, mentre la quantizzazione rende discreti i valori che può assumere il segnale. Infine, il processo di codifica converte la sequenza numerica in un flusso di bit.

I dispositivi utilizzati per le successive operazioni sul segnale registrato sono gli standard personal computer, ma anche laptops, subnotebook, grazie ancora al fatto che i passi in avanti della tecnologia nell'ambito informatico forniscono soluzioni sempre più piccole e trasportabili ma con prestazioni sempre più elevate. La gestione delle diverse operazioni (esecuzione del task, elaborazione e feedback del sistema) avviene mediante l'implementazione di pacchetti già pronti, molti dei quali reperibili in internet e di tipo *open source*. Per la gestione dei protocolli BCI il framework più usato è BCI2000 [35].

1.3.2 Descrizione delle tecniche di pre-processing del segnale

La misura del segnale EEG può essere affetta da problematiche di rumore. Infatti è possibile acquisire non solo il segnale di interesse, ma anche componenti indesiderate denominate rumore o artefatti. L'origine di questi artefatti può essere fisiologica e non. Per esempio, gli artefatti non fisiologici possono essere originati dai movimenti degli elettrodi, dalla strumentazione EEG o possono essere dei disturbi esterni (es. interferenze elettromagnetiche) che si accoppiano al segnale utile. In genere possono essere evitati tramite appropriate tecniche di schermatura per le apparecchiature ed un opportuno stadio di filtraggio. È importante ricordare che però il filtraggio deve avere una frequenza di taglio ottima proprio perché la banda in frequenza della componente d'interesse del segnale si sovrappone a quella del rumore. Quindi, eliminato il rumore, si può anche eliminare parte del contenuto informativo del segnale.

Per quanto concerne gli artefatti fisiologici, questi sono dei segnali elettrici che hanno origine dal paziente. Essi sono più difficili da eliminare. Questi includono potenziali muscolari e potenziali provenienti dagli occhi (possono essere di tipo corneoretinale e elettroretinale). Inoltre la sudorazione può alterare l'impedenza degli elettrodi e causare

artefatti nei segnali EEG. L'artefatto di maggiore interesse è quello oculare (EOG), il quale è facilmente riconoscibile nel segnale. L'EOG ha infatti un'ampiezza superiore a quella del segnale EEG. Per questo, una fra le tecniche più utilizzate per la rimozione dell'artefatto oculare è la sogliatura. Questa si basa sull'eliminazione della traccia con ampiezza superiore ad una certa soglia. Sono anche presenti altre tecniche in grado di rimuovere l'artefatto senza però eliminare la traccia. In [13] sono elencate alcune tecniche di elaborazione in quest'ottica. Si tende così ad "aggiustare" il segnale minimizzando le modifiche alla componente d'interesse.

Quanto descritto è la fase preliminare di elaborazione il cui fine è quello di ripulire il segnale dal rumore. Il passo successivo è invece l'applicazione di una metodica in grado di evidenziare unicamente la componente d'interesse, ottenendo una stima funzionale alle successive operazioni/analisi.

In generale, la tecnica più comune per l'analisi dell'EEG è l'*averaging convenzionale* [24]. Questa tecnica prevede di, dato un numero elevato di tracce, eseguire la media rispetto al tempo in modo da mettere in risalto la componente d'interesse ed attenuare il rumore di fondo. Per esempio, le tracce mostrate in figura 1.2.2, sono state ottenute proprio come media delle sweeps acquisite.

Questa operazione è possibile se sono valide specifiche assunzioni [24]. La prima prevede che il potenziale evento-correlato sia stazionario durante tutta l'acquisizione e quindi perfettamente correlato allo stimolo che lo genera. La seconda prevede che il segnale EEG registrato sia rappresentabile come rumore stocastico. Queste ipotesi sono valide per piccoli tratti di EEG ed infatti le tracce che si visualizzano sono parti di EEG per lo più inferiori ai due secondi di acquisizione. I limiti di questa tecnica sono evidenti. Se da un lato queste ipotesi non valgono in senso stretto, dall'altro mediare su un numero elevato di sweeps porta a perdere delle importanti informazioni cliniche, come nel caso della P300, in cui la latenza ed ampiezza hanno funzione diagnostica. Informazioni di questo tipo non sono ottenibili tramite averaging convenzionale. Inoltre, nel caso BCI, non è pensabile l'applicazione di tecniche di averaging, proprio perché ad ogni trial corrisponde una sweep e dall'informazione contenuta in quell'unica sweep bisogna pilotare l'output del sistema. Si parla così di tecniche *single trial* [44], ovvero tecniche che elaborano la parte di EEG corrispondente al task realizzato. Le metodiche utilizzate sono diverse e si distinguono innanzitutto nel dominio in cui lavorano: nel tempo, si hanno metodiche come modelli autoregressivi; nel dominio dello spazio, le tecniche più comuni sono Principal Component Analysis (PCA), Independent Component Analysis (ICA); tecniche nel dominio dello spazio e del tempo, come tecniche di coerenza. Questi sono solo alcuni esempi delle diverse tecniche presenti in letteratura.

1.3.3 Descrizione del processo di estrazione delle features

Il secondo blocco è quello della *features extraction* [23]. Dal segnale elaborato, è possibile estrarre una serie di parametri, i quali contengono l'informazione "in codice" della volontà dell'utente. L'estrazione delle features ha anche il compito di ridurre il numero di dati che verranno inviati al classificatore, riducendo così il time processing del sistema BCI. Infatti,

il numero di features calcolato è limitato e comunque inferiore al numero di campioni del segnale EEG.

Le tipologie di features da estrarre sono diverse. Si definiscono features calcolate nel dominio del tempo, nel dominio della frequenza e nel dominio tempo-frequenza. Esempi nel dominio del tempo sono il valore massimo e minimo di ampiezza del segnale, l'istante di occorrenza della componente d'interesse, l'area sotto la curva in un determinato intervallo d'interesse e alcuni parametri statistici come la correlazione. Esempi nel dominio della frequenza sono le trasformate di Fourier ed anche intervalli e picchi di potenza a determinate frequenze. Nel dominio tempo-frequenza le più diffuse sono le trasformate wavelet. Indipendentemente dalla natura delle features, lo scopo è il medesimo. L'obiettivo è quello di identificare delle peculiarità del segnale dipendenti dallo specifico task, ovvero lo specifico stimolo proposto al soggetto. Per esempio, nel caso degli *SCP*, tali parametri devono essere in grado di scindere nettamente il segnale relativo allo stimolo in alto rispetto a quello in basso. Ci si aspetta così che le features calcolate rispetto al segnale rappresentante lo stimolo in alto siano nettamente differenti rispetto a quelle calcolate per lo stimolo in basso.

1.3.4 Descrizione della procedura di classificazione

L'ultimo blocco è la classificazione e traduzione dell'output. Le tipologie di classificatori sono molteplici [23]: dalle metodiche lineari, come i *linear discriminant analysis* e *perceptroni*; metodiche non lineari, come *Support Vector Machine* (SVM) e reti neurali. Il blocco di classificazione ha però due diverse fasi di applicazione, correlate a due diverse fasi di funzionamento del classificatore, la fase off-line e la fase on-line. Nella fase off-line, l'obiettivo è quello di addestrare il classificatore sulla base dei dati acquisiti in precedenza. Quindi si esegue una fase di training, parte in cui si trova la combinazione ottima delle features tali da minimizzare l'errore di classificazione ed ottimizzare le performance. Una volta che si ha a disposizione un classificatore, si passa alla sua applicazione, ovvero la fase on-line. In questo caso, il classificatore assocerà al set di features calcolato per la sweep (parte di EEG in corrispondenza dello stimolo) un'etichetta rappresentante la classe di appartenenza ed in funzione di ciò fornisce l'output del sistema.

Il classificatore è strettamente legato al soggetto che esegue la prova, quindi non è possibile che classificatori addestrati sulle tracce acquisite da un altro individuo vengano utilizzati su altri soggetti. Proprio per questo motivo, il protocollo sperimentale deve prevedere una prima fase di semplice acquisizione del segnale EEG, senza che quindi il sistema BCI sia in funzione. Successivamente i dati vengono elaborati, come spiegato nei precedenti paragrafi, ed infine addestrato il classificatore (fase off-line). Da qui in poi sarà possibile per il soggetto effettuare delle prove on-line.

1.4 Indici di validazione del sistema BCI

Una volta che sono state scelte le tecniche di pre-elaborazione del segnale, il numero e tipo di features e quindi il classificatore, il sistema BCI viene così implementato. E' impor-

tante comprendere però quanto sia performante il sistema così costruito. Per questo sono stati introdotti dei parametri di valutazione specifici [5]. Si parla di *High Information Transfer Rate* [5], cioè la valutazione di quanto velocemente è in grado il sistema BCI di trasferire le informazioni. Questo aspetto è quindi implicitamente legato all'accuratezza del sistema nonché al tempo intercorso fra la stimolazione e l'output dello stesso. Altro parametro è *Ease-of Use* [5]. Anche se le tecnologie del sistema possono essere complesse, è necessario che comunque il dispositivo sia semplice ed immediato nell'utilizzo, riducendo al minimo le operazioni da compiere. La *Robustness* [5] è la possibilità di adattamento del dispositivo in diversi ambienti di applicazione.

Questi parametri di valutazione sono fra i più importanti introdotti, ma chiaramente, in funzione dello specifico sistema si possono implementare diversi criteri di valutazione. Per esempio, l'errore di classificazione di uno specifico stimolo. Questo errore viene solitamente calcolato rispetto a due dataset differenti. In generale, il set dei dati a disposizione viene suddiviso in due parti. Il primo insieme è chiamato *learning test*. I dati destinati a questo insieme sono impiegati nel processo di apprendimento del classificatore. Il secondo set di dati, in numero inferiore al learning set, è detto *testing set*. I dati in esso contenuti non vengono coinvolti nel training del classificatore. Le tracce che costituiscono i due dataset hanno classe di appartenenza nota. Quindi applicando il classificatore sia al learning che al testing set è possibile valutare se il risultato della classificazione è corretto o meno (errore di classificazione). Gli errori del testing set sono quelli che hanno valenza maggiore perché costituiscono una sorta di simulazione on-line del classificatore. Mentre infatti il classificatore ha già avuto conoscenza delle tracce di learning proprio per la fase di training, tali errori sono in genere più bassi rispetto al testing set. I dati del testing set sono invece sconosciuti al classificatore, il quale deve fornirne l'etichetta di appartenenza sulla base dell'informazione raccolta nel learning set.

1.5 Brain Computer Interface basata su P300

Fra tutte le tipologie di sistemi BCI introdotti in precedenza nel paragrafo 1, si considera la *P300 based BCI*, ovvero il sistema BCI che utilizza come segnale di controllo la componente P300, un potenziale evento correlato. Secondo le definizioni date, questo sistema è NON INVASIVO, perché basato sull'acquisizione del segnale EEG, ed INDIPENDENTE, in quanto la componente P300 è innata nell'uomo. A differenza delle BCI basate sui SCPs oppure ERD, le quali hanno bisogno di un periodo più o meno lungo per l'addestramento del paziente in modo da potenziare la propria risposta, la BCI su P300 non ha alcuna necessità di addestramento.

Per quanto detto prima, lo schema a blocchi di figura 1.3.1 rimane ancora valido.

1.5.1 Analisi dei processi di generazione dell'onda P300

Se un soggetto è sottoposto ad un generico protocollo di stimolazione e contemporaneamente viene registrato il segnale EEG, allora è possibile riconoscere nel tracciato acquisito delle componenti chiamate *potenziali evocati* (PE) [11]. I PE appaiono come deflessioni monofasiche e riflettono variazioni sincrone di lenti potenziali post-sinaptici dei neuroni corticali.

Inoltre, tali potenziali sono dipendenti dal tempo di stimolazione e possono essere più o meno riproducibili in similari condizioni di esperimenti.

Un generico potenziale evocato viene indicato con la relativa polarità e latenza. Per esempio, N100 indica una deflessione negativa (*N* sta appunto per negativo) che occorre 100 ms dopo lo stimolo.

Un particolare sottogruppo dei potenziali evocati sono i *potenziali cognitivi*.

La definizione di *potenziali cognitivi* [16] deriva dal generico rapporto che presentano con l'attività mentale del soggetto che ha luogo come reazione a stimoli in arrivo in uno o più canali sensoriali. Questi potenziali ricevono anche l'attributo di *evento correlati* perché gli stimoli fisici che li determinano nel paradigma sperimentale assumono il significato di *evento* (ERPs, dall'inglese Event Related Potentials), equivalente al riconoscimento di un evento target all'interno di una serie di stimoli in sequenza. Questa definizione nasce proprio dal primo studio condotto in letteratura sulla componente endogena P300 a cura del dottore Sutton nel 1965.

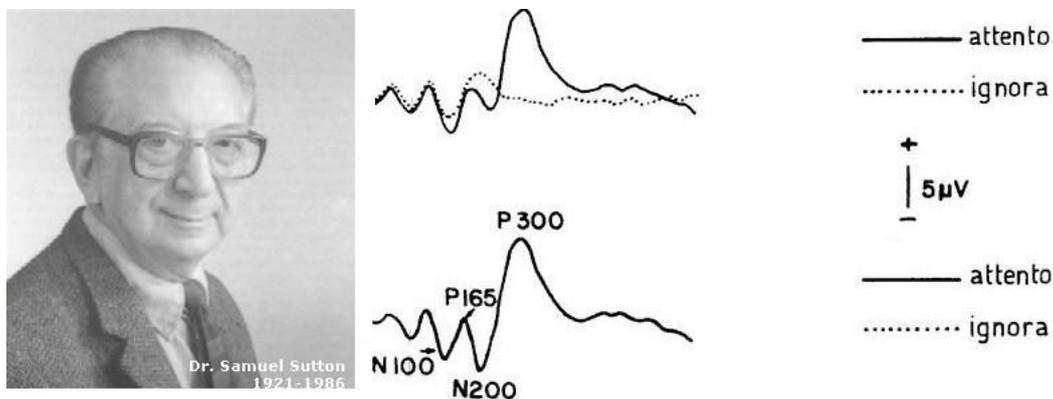


Figura 1.5.1: S. Sutton [s9] (sx) e l'onda P300 [16](dx).

La componente P300 è un potenziale evocato evento correlato, con polarità positiva che si manifesta con un picco a circa 300 ms dopo lo stimolo target. È una componente *endogena*, cioè il potenziale viene elicitato da meccanismi interni del cervello in risposta ad uno stimolo esterno (si definisce *esogeno* il potenziale che è elicitato in funzione delle caratteristiche fisiche dello stimolo esterno, come per esempio l'intensità, la durata e la frequenza dello stimolo stesso. Un potenziale di questo tipo è per esempio SSVEP di figura 1.2.4).

Ormai il termine P300 è considerato sinonimo di tutte le componenti endogene, anche se in realtà, la risposta allo stimolo è caratterizzata anche da altre componenti che precedono la P300, la cui natura non è unicamente endogena. Studi condotti [16], hanno dimostrato che le componenti presenti tra i 50 ms e 80 ms che seguono un transitorio di accensione dello stimolo sono principalmente esogene. Le componenti con latenza compresa fra 80 ms e 200 ms, dopo lo stimolo, hanno caratteristiche sia endogene che esogene. Infine, hanno natura prevalentemente endogena quelle con latenza superiore a 200 ms dopo lo stimolo.

In riferimento alla figura 1.5.1 (dx), prima dell'occorrenza della P300, si evincono altri picchi significativi. In particolare, la prima componente [16] è N100, un picco negativo a circa 100 ms dopo lo stimolo, seguita dalla P165, un'onda vertice-positiva con latenza pari a circa 160 ms dopo lo stimolo. Per quanto detto sopra, entrambe le componenti hanno caratteristiche endogene-esogene.

Il complesso successivo di potenziali endogeni N200-P300 comprende una componente negativa a 200 ms dopo lo stimolo, seguita dal picco positivo a 300 ms dopo lo stimolo. Quest'ultimo è appunto l'onda P300.

Come ogni altro potenziale evocato, la componente P300 [31] è tradizionalmente collegata a processi di riconoscimento, alla capacità di prendere decisioni, all'attenzione ed all'aggiornamento della memoria. Per questo è largamente utilizzata in ambito clinico, soprattutto per lo studio di malattie cognitive come la depressione, la schizofrenia e la demenza, in cui si controlla la risposta della P300, confrontandola con quella dei soggetti sani [11].

La componente P300 è misurata attraverso la valutazione della sua latenza e ampiezza. Si definisce [31] *latenza* (ms) il tempo intercorso fra lo stimolo e il punto di massima ampiezza positiva all'interno della finestra temporale considerata. La latenza è più corta nell'area frontale e diventa più lunga nell'area parietale dello scalpo [31, 32]. E' un indice della velocità di classificazione, la quale è proporzionale al tempo richiesto per identificare lo stimolo e riconoscere l'evento atteso. Infatti, molti test neurofisiologici producono la più alta correlazione fra la latenza della P300 e la capacità cognitiva di quanto rapidamente i soggetti allocano le proprie risorse attenzionali [31]. Si può comprendere come la latenza ha una valenza clinica notevole, sia per le indagini nell'evoluzione dell'essere umano (nell'età dello sviluppo, la latenza della P300 diminuisce, mentre ritorna entro range "normali" col passare dell'età [31]), sia per l'identificazione dei diversi stadi delle patologie (per esempio l'incremento del livello della demenza provoca un aumento nella latenza della P300 [31]).

Si definisce [31] *ampiezza* (μV) la differenza fra la media del pre-stimolo e il più grande positivo picco nell'onda ERP nella finestra temporale fra i 250 ms e 500 ms dopo lo stimolo, anche se la finestra temporale può subire variazioni in dipendenza dalla modalità di stimolazione, dalle condizioni dell'evento proposto e da una serie di variabili strettamente collegate al soggetto. La distribuzione dell'ampiezza della P300 cambia lungo gli elettrodi mediani (Fz,Cz,Pz) con un incremento dalla zona frontale alla zona parietale [31, 32].

Studi condotti [31, 32, 16] hanno evidenziato che la P300 consta di due componenti:

1. *P3a*, la quale ha la massima ampiezza nella parte centrale e parietale dello scalpo ed una latenza molto corta, a circa 200 ms dopo lo stimolo. Inoltre, si verifica anche quando il soggetto rileva la differenza fra lo stimolo atteso e quello frequente, ma non è istruito ad eseguire alcun task;
2. *P3b*, dipendente dalla rilevanza del task proposto al soggetto. Le caratteristiche di latenza e ampiezza sono uguali a quelle precedentemente elencate.

Rimangono così da indagare le origini neurali di queste componenti [31, 16]. La componente P3a sembra essere correlata al meccanismo di attenzione del soggetto durante l'occorrenza degli stimoli; mentre, la P3b sembra presentarsi quando le risorse attenzionali lasciano spazio alle operazioni di memoria nella zona temporale-parietale inerenti al riconoscimento del target. Nella figura 1.5.2 si presenta uno schema di quanto detto ed è da notare la sommazione fra la componente P3a tratteggiata e la P3b in linea continua.

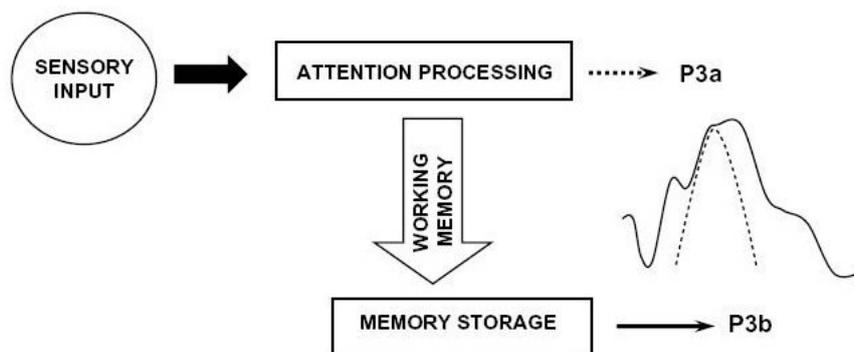


Figura 1.5.2: *P3a e P3b* [31].

Descritte le proprietà della P300, ci si sofferma adesso sul paradigma in grado di elicitarla. Il metodo più diffuso è sicuramente il paradigma *oddball* [10], che, nel caso più semplice, consiste nella presentazione di due differenti stimoli distribuiti in modo pseudo-random in una sequenza. Uno di questi si presenta frequentemente ed è detto stimolo *standard*, l'altro, invece, è l'evento *target*, che appare raramente nella sequenza di stimoli ed in modo inaspettato. Gli stimoli standard e target possono essere suoni a frequenze differenti, figure con diversi colori, frecce che si illuminano e altro ancora.

In figura 1.5.3 è rappresentato schematicamente ciò che succede nel paradigma oddball [31].

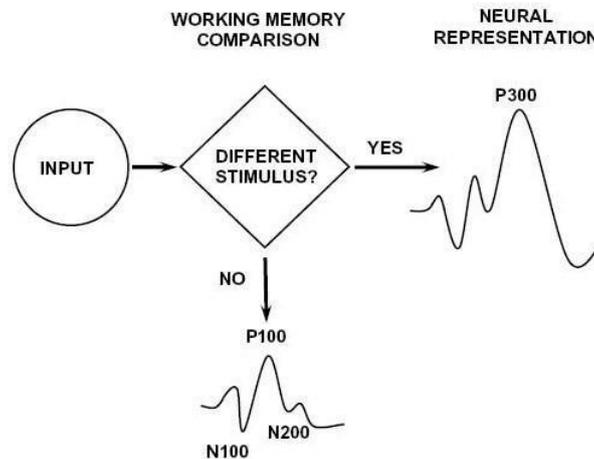


Figura 1.5.3: *Paradigma Oddball* [31].

In corrispondenza dell'*input*, che rappresenta lo stimolo proposto al soggetto, possono verificarsi due condizioni: se l'evento corrente è uno stimolo *standard*, caso NO in figura 1.5.3, viene prodotto un semplice potenziale evocato sensoriale con i complessi P100-N200 caratteristici. Se invece lo stimolo corrente è *target*, caso SI in figura, allora viene generata la P300, il potenziale evento-correlato [31, 11].

1.5.2 Descrizione dell'Interfaccia Grafica per l'elicitazione della P300 nei sistemi BCI

La riflessione che propone il paragrafo precedente è proprio quella per cui se si vuole creare un sistema BCI con segnale di controllo la P300, accanto allo schema a blocchi di figura 1.3.1 va inserita un'altra componente fondamentale dello stesso sistema: l'interfaccia grafica per la stimolazione del soggetto che esegue l'esperimento, la quale deve essere in grado di elicitarne la componente P300.

Il sistema BCI basato sulla P300 è stato introdotto da Farwell e Donchin nel 1988 ed è chiamato *P300-speller* [12].

Il paradigma implementato permette la scrittura a video di parole. Il soggetto viene posizionato di fronte ad un monitor e mostrata la parola o frase da comporre. Successivamente, è presentata una matrice 6x6 di caratteri. Le righe e le colonne vengono illuminate in modo random durante l'esperimento (vedi figura 1.5.4) [25]. L'utente dovrà porre la sua attenzione sul carattere d'interesse all'interno della matrice. Quando viene illuminato proprio il carattere che serve per la composizione della parola, sarà elicitata la componente P300. Questo procedimento verrà ripetuto fino a che la parola scelta sarà composta.

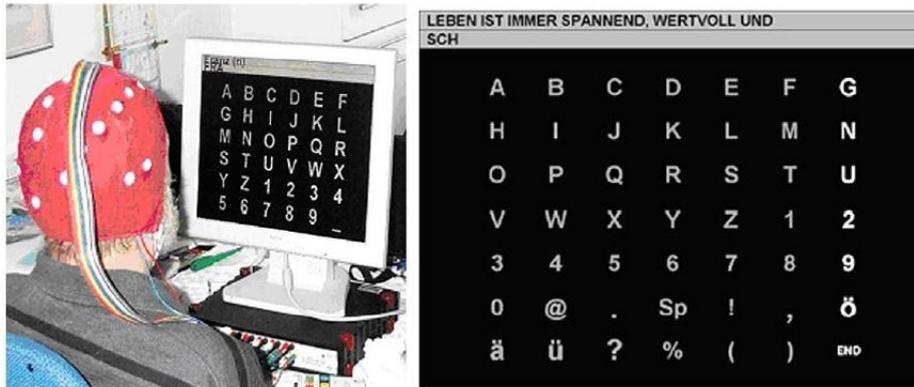


Figura 1.5.4: *Interfaccia grafica della P300 speller [25].*

I pazienti ALS sottoposti a questo esperimento hanno però dimostrato delle difficoltà a comunicare attraverso lo spelling carattere per carattere. Si tratta di un aspetto clinico, inerente alla dimensione della matrice 6x6 a cui il soggetto doveva prestare attenzione. Per focalizzare l'attenzione verso un carattere, il soggetto tende involontariamente a muovere gli occhi verso la direzione d'interesse e questo può comportare un grosso limite [25, 34]. Per questo motivo, Sellers e Donchin [34] hanno introdotto nel 2006 un paradigma a quattro stimoli in sequenza: YES,NO,PASS,END. E' un modo di comunicazione alternativo, limitato per il numero di scelte proposte ma sicuramente immediato da un punto di vista pratico. Inoltre, verificano anche un aspetto importante ancora non esplicitamente messo in luce ed inerente al tipo di stimolazione. Nel caso della P300 speller, lo stimolo è chiaramente di tipo visivo. Se invece si introducono dei paradigmi su più scelte, anche il modo di presentare queste viene modificato.

Si possono avere, così, modalità di stimolazione diverse. La stimolazione di tipo visivo [25, 34] consiste nella presentazione di immagini. Queste possono far riferimento a campi semantici variegati: dall'ambito prettamente culinario, a quello medico oppure possono essere immagini di semplice comunicazione SI o NO, come indicato in [34]. La stimolazione di tipo uditivo [34] consiste, invece, nella presentazione di toni a frequenze differenti per i target e non target. E' anche possibile associare ad una specifica immagine anche uno specifico suono, ottenendo una stimolazione visiva-uditiva [34]. Inoltre, per aumentare l'attenzione del soggetto, è possibile assegnargli un ulteriore compito, come per esempio contare il numero di eventi target presentatesi oppure premere un pulsante in corrispondenza di questi. In questo caso si parla di *oddball attivo* [11], mentre per gli altri casi si parla di *oddball passivo* [11].

E' importante sottolineare che i paradigmi non sono tutti uguali e che la componente P300 può subire notevoli variazioni in termini di ampiezza e latenza proprio in funzione del paradigma utilizzato. Per esempio, in [34] non si sono rilevate importanti differenze nella P300 rispetto al paradigma visivo piuttosto che uditivo; mentre un significativo aumento dell'ampiezza si è riscontrato quando gli stimoli erano sia visivi che acustici. Altra variabilità, per una sequenza di stimoli, è indotta dal TTI [31], ovvero dal *target*

-to-target interval, definito come il numero di non target stimoli (standard) che occorrono prima dell'identificazione dello stimolo target. Per esempio, piccoli TTI generano una componente P300 più corta rispetto a valori più alti di TTI, mentre valori di TTI fra 6-8 secondi eliminano definitivamente questa correlazione [31].

In [17], a partire dall'idea che il rapporto segnale-disturbo della componente P300 aumenta con l'infrequenza dello stimolo target, è stata condotta un'analisi su quale sia il miglior rapporto fra numero di stimoli target e quelli standard, proponendo un rapporto di 1:8, cioè ogni 9 stimoli, uno deve essere target, gli altri otto sono stimoli non target. Chiaramente, tutte le possibili soluzioni per migliorare l'acquisizione del segnale devono far fronte ad un altro problema: il tempo. Per esempio, se è vero che il rapporto 1 : 8 migliora l'SNR del segnale, questo comporta l'aumento del numero di stimoli per ciascuna scelta e quindi aumenta il tempo necessario per raggiungere il target. Stesse considerazioni valgono per TTI.

Oltre a queste variabili legate al protocollo (numero di stimoli, intensità e posizione degli stessi nella sequenza), un fenomeno da non trascurare è quello dell'*abituazione* [18, 33]. Come si è già detto, all'interno del paradigma oddball lo stimolo viene ripetuto più e più volte. Questo può indurre un processo di apprendimento che ha come conseguenza la riduzione dell'ampiezza della P300 [18, 33].

1.6 Conclusioni

Questo capitolo riporta una descrizione del funzionamento di un generico Brain-Computer Interface, indicando quali sono i principali user nonché dispositivi controllati. In particolare, i dati di questa tesi sono relativi a pazienti ALS.

Il sistema BCI più diffuso è quello basato sul segnale EEG ed, in particolare, il sistema BCI basato sulla P300 perchè quello che consente un utilizzo immediato del dispositivo senza dover fare delle prime fasi di addestramento del segnale di controllo. Lo schema a blocchi è costituito dalla parte di acquisizione del segnale, dalla parte di pre-processing del segnale registrato, dalla parte di features extraction ed infine dalla parte di applicazione del classificatore, con la successiva traduzione del risultato in feedback.

1.7 Scopo della tesi

Questo lavoro è svolto in collaborazione con l'istituto IRCCS (Istituto di Ricovero e Cura a Carattere Scientifico) del San Camillo (Venezia/Lido), in cui è presente il sistema BCI basato sulla P300, utilizzato per le prove condotte in questa tesi.

La prima parte del lavoro è volta alla comprensione del funzionamento del sistema BCI del San Camillo e, per questo, si è svolta principalmente in questa struttura. Si è avuta la possibilità di guardare da vicino ogni singolo step: da come il segnale viene acquisito e memorizzato, come questo viene elaborato e quindi il feedback fornito per mezzo del classificatore. Tutto ciò verrà descritto nel dettaglio nel Capitolo 2.

La parte sperimentale, cuore del lavoro di tesi, in una validazione off-line di due tecniche bayesiane per la stima single trial di potenziali ERP. Lo scopo è quello di valutare la convenienza dell'introduzione di queste tecniche quali procedure di pre-processing da applicare ai segnali prima della classificazione. Nella validazione off-line gli algoritmi vengono testati su un set di dati acquisito in altre sessioni quindi senza che il sistema BCI sia in funzione con l'acquisizione on-line del segnale EEG. Questo approccio è doveroso per ogni nuova tecnica. Se le prove sono soddisfacenti off-line, allora la metodologia può essere candidata alla validazione on-line, in cui l'algoritmo verrà testato proprio all'interno di un protocollo sperimentale con diversi soggetti, sani e patologici.

Lo scopo della tesi è valutare l'effetto di un pre-processing bayesiano sulle performance di un sistema BCI basato sulla P300. La prima tecnica valutata, indicata con l'acronimo B2S (*bayesian two step*), utilizza un approccio a due passi: il primo consiste nel calcolo del profilo atteso del potenziale evocato, mentre il secondo nella stima vera e propria single trial.

La seconda tecnica utilizza un approccio di Multi Task Learning (MTL), che tratta l'insieme delle sweeps, di cui si vuole fornire la stima single-trial, come una *popolazione omogenea*, composta da elementi in qualche modo simili fra di loro. L'algoritmo fornisce in output un segnale medio, caratteristico dell'intera popolazione e le singole stime.

Le tecniche saranno testate sia sui dati simulati che sui dati reali. La prima analisi riguarderà le stime ottenute. Per i dati simulati sarà possibile confrontare direttamente il segnale vero con quanto stimato dai due metodi B2S ed MTL. Mentre per i dati reali si valuterà quanto i metodi B2S ed MTL siano in grado di fornire una stima della componente ERP, valutata come una deflessione positiva a circa 300 ms.

Successivamente, si addestreranno i classificatori sulla base delle features estratte dalle stime con i due metodi B2S ed MTL. Infine, verranno riportati gli errori di classificazione ottenuti. In particolare, nel caso dei dati reali, tali risultati verranno confrontati con le performance del sistema BCI attualmente in uso al San Camillo.

Capitolo 2

Il sistema Brain Computer Interface dell'IRCCS San Camillo

In questo capitolo si analizzerà nel dettaglio il sistema di Brain-Computer Interface basato sulla P300 dell'IRCCS San Camillo che è stato usato per raccogliere i dati che verranno analizzati nel proseguo di questa tesi.

2.1 Il Paradigma Oddball

Il paradigma utilizzato dagli studiosi dell'IRCCS San Camillo prevede che al soggetto sotto esame vengano presentati su di un monitor un'interfaccia del tipo di quelle mostrate in figura 2.1.1.



Figura 2.1.1: *Interfaccia utilizzata nel sistema BCI del San Camillo [39]. (A) Se il target è indicato dalla freccia LEFT, in corrispondenza dello stimolo visivo (B) il soggetto riconosce l'evento target, si elicitava la componente P300, la quale viene rilevata dal classificatore, generando in output lo spostamento del cursore verso lo stimolo (C).*

Si tratta di un paradigma a quattro scelte, le quali rappresentano quattro possibili bisogni: “ho fame”, “voglio una mela”, “ho bisogno di un dottore”, “voglio dormire”. Ad ognuna di queste è associata una direzione (RIGHT, LEFT, UP, DOWN), rappresentata dalla corrispondente freccia. Il cursore blu è posto al centro dello schermo (A) e dovrà essere pilotato dal soggetto verso la direzione della propria scelta.

All'inizio di ogni sessione, viene scelto l'obiettivo da raggiungere, che equivale a definire il target del paradigma oddball, in modo tale che il soggetto concentri la sua attenzione su tale stimolo, trascurando gli altri.

In riferimento alla figura 2.1.1, il target è individuato dalla freccia LEFT. Quando tale freccia si illumina (B), la P300 viene elicitata, il sistema BCI ha il compito di riconoscerlo e di fornire come output il movimento del cursore verso lo stimolo (C) [39].

Il cursore dista quattro passi discreti dai possibili stimoli e i suoi movimenti sono unicamente verticali od orizzontali perché riferiti alle quattro frecce posizionate agli apici di una croce. Quindi, nessun movimento in direzione obliqua è permesso.

La tipologia di stimolazione è solo visiva: quando viene proposto uno stimolo luminoso, la freccia corrispondente si illumina.

Inoltre, viene chiesto agli utenti, tranne a coloro con maggiori disabilità, di schiacciare un pulsante in corrispondenza dell'evento target. Ciò non ha alcuna funzione all'interno del sistema BCI se non quella di aiutare l'utente a mantenere la concentrazione [39].

L'illuminazione della freccia ha una durata di 150 ms e l'intervallo tra l'illuminazione di una freccia ed un'altra è pari a 2.5 secondi, in modo da privilegiare l'applicazione *on-line* del sistema. Infatti, proprio nel caso on-line, questo è il tempo che al massimo dovrà impiegare il sistema per elaborare la traccia acquisita e fornire il feedback visivo all'utente [39].

Le frecce si illuminano secondo uno schema random a blocchi, in cui ogni blocco presenta quattro accensioni delle frecce con ordine casuale. Il rapporto target-non target è 1:3. Ogni quattro stimoli, uno è l'evento target, gli altri tre sono gli eventi standard. In termini di probabilità, l'evento target è al 25%, gli stimoli standard al 75%. Si definisce così *sessione*, l'insieme dei trial necessari al raggiungimento di un target [39].

2.2 Schema a blocchi del sistema

Di seguito verranno descritti i blocchi fondamentali del sistema BCI del San Camillo secondo lo schema 1.3.1 del capitolo 1. In particolare, tale schema viene adattato in due particolari fasi di funzionamento del sistema BCI, quella off-line e quella on-line. Nella fase off-line si svolgono tutte quelle operazioni necessarie a preparare il classificatore, utilizzando i dati acquisiti in precedenti sessioni. Quindi lo schema a blocchi 1.3.1 si particolarizza come indicato in figura 2.2.1.

In fase off-line, il blocco di pre-processing del segnale viene applicato direttamente al dataset (insieme di sweeps target e non target acquisite in precedenti sessioni), in modo tale che l'output del blocco di classificazione sia il classificatore SVM (Support Vector Machine) addestrato.

Nella fase on-line invece rimane valido lo schema 1.3.1, in cui i blocchi di elaborazione adesso vengono applicati alla singola sweep estratta in corrispondenza dello stimolo. Il blocco di classificazione applicherà il classificatore addestrato off-line, in modo da determinare l'etichetta TG/NTG da associare alla sweep corrente e generare il corrispondente feedback (movimento del cursore di un passo discreto).

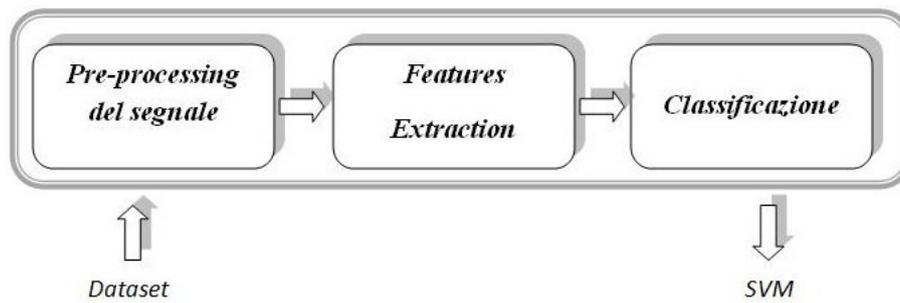


Figura 2.2.1: Schema a blocchi in fase off-line.

Nei successivi paragrafi, meno che nel caso di acquisizione del segnale, verrà così specificato la diversa applicazione dei blocchi nel caso off-line ed on-line.

2.2.1 Blocco di acquisizione del segnale EEG

Per l'acquisizione del segnale EEG [39, 28], si utilizzano degli elettrodi a “coppetta” in argento rivestiti in cloruro di argento (Ag/AgCl) sintetizzati del diametro di circa 10 mm. Sono elettrodi waterproof, cioè lavabili e riutilizzabili e vengono connessi mediante dei cavi ad un amplificatore Neuroscan.

Il protocollo di acquisizione prevede l'uso di quattro elettrodi posizionati rispettivamente nella zona frontale, centrale, parietale e occipitale della parte mediana dell'emisfero [28, 39]. Riproponendo il Sistema Internazionale 10-20 [20], si fa riferimento alle zone Fz,Cz,Pz,Oz. Questi hanno come riferimento il lobo dell'orecchio, sito elettricamente neutro.

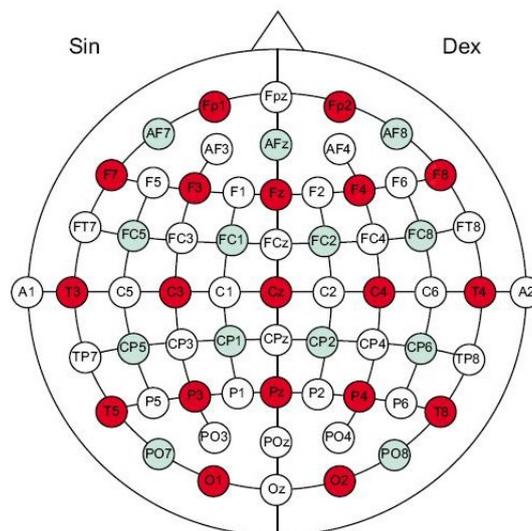


Figura 2.2.2: Sistema Internazionale 10/20 [s13].

Contemporaneamente all'attività elettrica cerebrale, vengono registrati anche i movimenti

oculari orizzontali e verticali (EOG). Il numero totale dei canali utilizzati è pari a cinque (Fz,Cz,Pz,Oz,EOG) [28, 39].

Il segnale EEG, acquisito secondo il protocollo descritto, verrà amplificato dal sistema SynAmps (NeuroSoft, Inc.). Questo esegue un filtraggio passabanda fra 0.15 e 30 Hz e la digitalizzazione con frequenza di campionamento a 200 Hz (risoluzione a 16 bit) [39, 28].

Il segnale EEG viene acquisito in continuo e, per l'analisi, ciò a cui si fa riferimento è la parte di EEG in corrispondenza dello stimolo luminoso.

La figura 2.2.3 chiarisce il contesto.

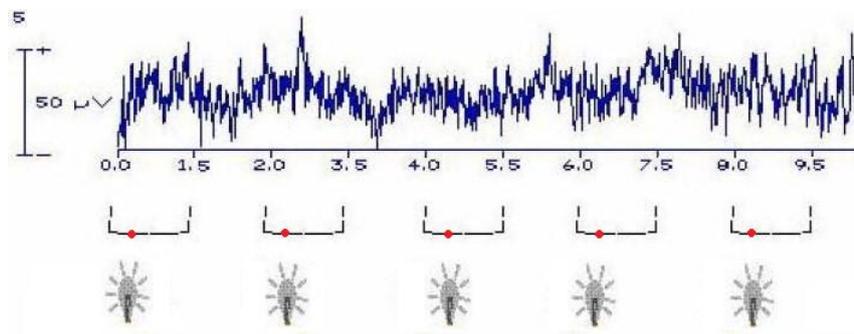


Figura 2.2.3: *EEG in continuo durante il paradigma oddball [14]. La parte EEG on-line che viene estratta è quella in corrispondenza dello stimolo, rappresentato dalla lampadina e istante di occorrenza in rosso. La sweep è così la parte di di segnale che va da 500 ms prima dello stimolo a 1000 ms dopo lo stimolo stesso.*

In blu è rappresentato il segnale EEG che viene registrato durante l'esecuzione del paradigma oddball. Le lampadine rappresentano invece gli istanti temporali in cui si presentano gli stimoli luminosi.

Si definisce *sweep* la parte di EEG compresa fra 500 ms prima lo stimolo ed un secondo dopo lo stimolo, rappresentata in figura 2.2.3 dal metà rettangolo.

Rispetto alla scala [-500, 1000] [ms], l'istante 0 ms è quello di stimolazione visiva.

Con frequenza di campionamento pari a 200 Hz, la sweep è costituita da un vettore pari a 300 campioni. In particolare, i campioni relativi ai cinque elettrodi vengono affiancati, per cui in corrispondenza di una sweep, si avrà una matrice 300x5. L'ordine degli elettrodi non è casuale ed è definito all'inizio del protocollo di acquisizione. Nel caso in esame, l'ordine delle colonne in termini di elettrodi è il seguente

Si definiscono target le sweeps in corrispondenza degli stimoli target; sweeps non target, in corrispondenza degli stimoli standard. In figura 2.2.4 si rappresenta il protocollo di acquisizione, cioè i quattro canali Fz,Cz,Pz,Oz, e le rispettive tracce che si registrano distinte per tipologia di stimoli.

In particolare, come chiaramente visto in figura 2.2.4, in corrispondenza delle sweeps target ci si aspetta di registrare una traccia in cui sia presente la componente P300 che è

INDICE COLONNA	ELETTRODO
1	Fz
2	Cz
3	Pz
4	EOG
5	Oz

Tabella 2.1: Ordine delle colonne della matrice dati in funzione degli elettrodi.

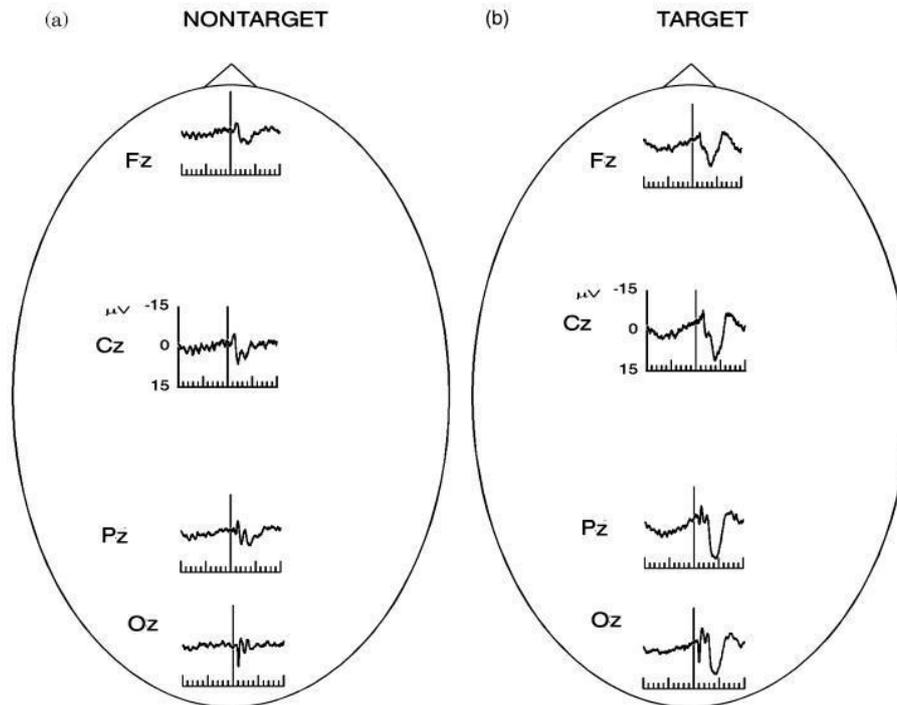


Figura 2.2.4: Tipico andamento del segnale EEG in corrispondenza di stimoli non target (a) e target (b)[11].

stata elicitata (b). In corrispondenza degli altri tre stimoli standard ci si aspetta di registrare sweeps con potenziali evocati visivi senza P300 (a).

Le sweeps registrate durante una sessione, una volta suddivise in target e non target, vengono salvate all'interno della matrice dati con l'acronimo rispettivamente di *ss2* ed *ss1*. Queste due matrici hanno dimensione pari a $300 \times (5 \times n2)$ e $300 \times (5 \times n1)$ rispettivamente, dove 300 è il numero di campioni nell'intervallo temporale da -500 ms a 1000 ms con frequenza di campionamento pari a 200 Hz, 5 è il numero degli elettrodi rispetto cui si acquisisce il segnale EEG, $n2$ è il numero di stimoli target ed $n1$ il numero di stimoli non target. Chiaramente, il numero di sweeps non target per elettrodo è superiore al numero di sweeps target, sempre per elettrodo.

2.2.2 Blocco di pre-elaborazione del segnale

Si inizia la descrizione del blocco di pre-elaborazione del segnale in fase off-line. Per prima cosa, il blocco di pre-processing applica alla matrice dati, sia target che non target, un filtraggio passa-basso, con frequenza di taglio a 15 Hz. Successivamente viene applicata la correzione per l'artefatto oculare mediante tecnica di sogliatura, per cui tutte le sweeps il cui valore è superiore in modulo a $100 \mu\text{V}$ vengono eliminate [39]. Fatto ciò, il passo successivo è quello di applicare un metodo automatico che determina la deflessione P300 più significativa rispetto ai canali di acquisizione, mediante una metodica *fuzzy* [39]. In questa fase la matrice dati utilizzata è quella target, di dimensione $300 \times (5 \times n^2)$. Per ogni canale verranno disposte sul segnale tre crocette che indicano la combinazione più significativa del potenziale evento-correlato P100-N200-P300. Per esempio, in figura 2.2.5 si mostra il risultato per il canale Pz.

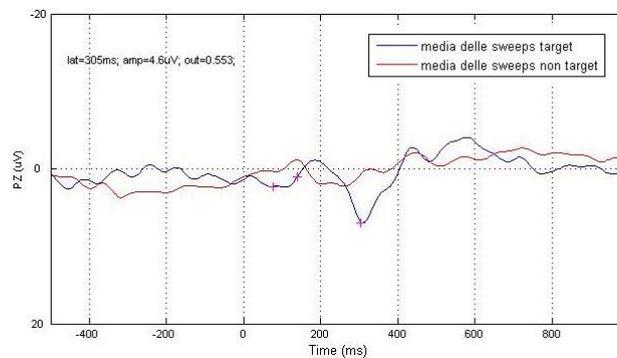


Figura 2.2.5: *Fuzzy System* rispetto al canale Pz. In rosso la media delle sweeps target, in blu la media delle sweeps non target.

In figura 2.2.5 si nota come le crocette indicano, in modo più o meno esatto, le tre componenti fondamentali del potenziale ERP. La combinazione più evidente viene selezionata in modo automatico rispetto ad un canale, ma l'utente ha comunque la possibilità di cambiarlo, selezionando quello che ha la combinazione che reputa più significativa.

Il successivo step consiste nel calcolo della componente ICA mediante il tool *FastICA*. Questa procedura ha l'obiettivo di selezionare la sorgente ottimale che genera il segnale acquisito, in modo che si possa ridurre di un fattore di cinque la matrice dati rispetto cui applicare i successivi passaggi. Tale sorgente viene scelta in modo automatico proprio sulla base del risultato del sistema fuzzy. Infatti verrà selezionata la sorgente che presenta la deflessione P300 più simile a quella identificata dal metodo fuzzy. Anche in questo caso, l'utente ha la possibilità di modificare la componente scelta automaticamente [39]. Si propone in figura 2.2.6 un esempio: in (a) è rappresentata la componente ICA selezionata, mentre in (b) la media delle componenti ICA sovrapposta alla componente selezionata. Da notare nel grafico le crocette determinate con il metodo fuzzy.

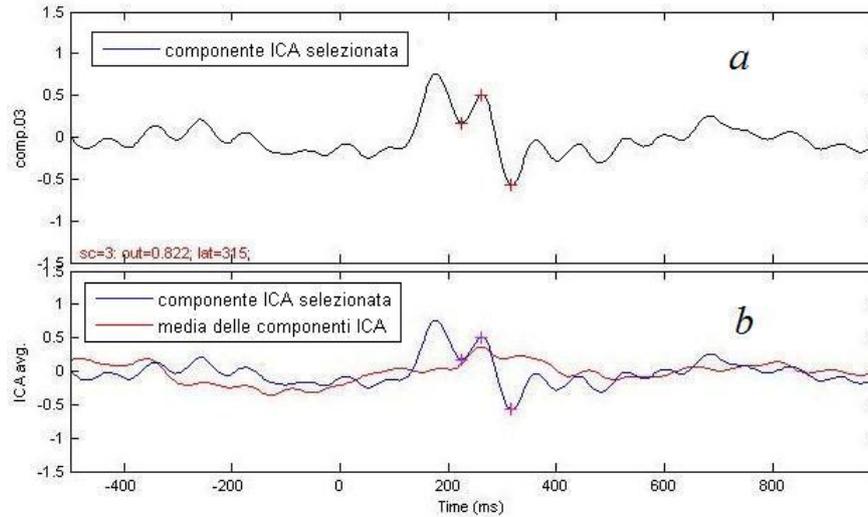


Figura 2.2.6: *Componente ICA selezionata (a). In (b) è mostrata la componente ICA selezionata (blu) e la media delle componenti ICA (rosso)*

Selezionata la componente ICA, tutte le sweep, sia target che non target, vengono moltiplicate per tale sorgente.

In fase on-line, le elaborazioni di pre-processing destinate alla sweep corrente sono il filtraggio a 15 Hz ed il prodotto di tale sweep con la sorgente ICA selezionata in fase off-line.

2.2.3 Blocco di features extraction

In fase off-line, una volta ottenuta la matrice dati “normalizzata” alla sorgente ICA sia per le sweeps target che non target, il passo successivo è l'estrazione delle features, il cui numero complessivo è 78 [6, 39].

Nel caso specifico, le features sono di diverso tipo e sono applicate indistintamente alle sweeps target e non target. Alcuni esempi sono la potenza in diversi punti dell'intervallo temporale, le trasformate wavelet, l'ampiezza e la latenza [6]. Lo scopo è quindi di valutare delle caratteristiche in grado di massimizzare la differenza fra le sweeps target e le non target. Ad esempio, è logico attendersi che, in caso di sweep target, l'ampiezza nell'intervallo fra i 250 ms ed i 600 ms sia in generale maggiore rispetto all'ampiezza della sweep non target. Infatti nel caso target dovrà essere presente la componente P300, nel caso non target invece no.

Si ricorda che la procedura prevede la suddivisione del dataset in due parti, il learning e testing set. Questo viene effettuato in questo punto dello schema. Infatti, le medesime features vengono calcolate per il learning e testing set. In particolare, si supponga che il numero complessivo di sweeps del learning set sia target che non target sia pari complessivamente ad N . La matrice delle features calcolate rispetto al learning set avrà dimensione $78 \times N$. Di tale matrice si calcolano il valore medio di ogni features e la standard deviation. Rispetto a tali valori vengono normalizzate le features del learning set

appena calcolate. Successivamente si calcolano le features del testing set e questi vengono normalizzati rispetto al valor medio e standard deviation del learning set.

In fase on-line, le features vengono calcolate rispetto la sweep corrente elaborata e si ottiene un vettore di 78 parametri. Questi vengono normalizzati al valor medio e standard deviation calcolati in fasi off-line ed inerenti al learning set.

2.2.4 Blocco di classificazione

In fase off-line, come già anticipato, il blocco classificazione crea il classificatore. Poiché sappiamo a priori la classe di appartenenza delle sweeps (target o non target), l'ambito è quello della classificazione supervisionata e il classificatore implementato è SVM (software *SVM^{light}* versione 5.00) [19]. I classificatori [39] vengono addestrati per mezzo di una funzione kernel che usa due parametri: *gamma* e *cost factor* [39]. Il primo ha un valore fisso impostato prima del training; il secondo invece viene scelto all'interno di una griglia di valori. Questo è il valore ottimo ottenuto dopo 20 iterazioni di cross-validazione, in cui si minimizza una funzione costo che tiene conto degli errori di classificazione delle sweeps sia target che non target [39]. Il classificatore viene così addestrato sulla base delle features calcolate in precedenza sul learning set.

In fase on-line, in funzione del vettore delle features calcolato rispetto alla sweep corrente, il classificatore SVM, addestrato off-line, dovrà fornire la classe di appartenenza (TG/NTG), in moda da generare il feedback appropriato.

2.3 Protocollo sperimentale applicato per l'acquisizione del data base

Prima di passare allo schema del protocollo sperimentale svolto, bisogna definire il funzionamento del sistema BCI in due diverse applicazioni: le sessioni di learning e quelle di testing.

- *sessione di learning*: è la fase di acquisizione del segnale EEG, senza però che questo venga elaborato dal sistema per fornire il reale feedback. Si parla cioè di *feedback ideale*. Il sistema BCI non sarà in funzione ed in corrispondenza di un evento target il cursore si sposterà automaticamente verso tale direzione, mentre nel caso degli stimoli standard non si avrà nessun movimento. Ogni sessione termina così con il raggiungimento del target e consta di quattro blocchi random in sequenza; mentre, in termini di stimoli, ci saranno da un minimo di 13 stimoli ad un massimo di 16 stimoli. In questa prima parte, il fine è quello di acquisire un numero opportuno di tracce in modo da poter creare il primo classificatore per il soggetto. In questo modo, la fase successiva è il funzionamento off-line del sistema BCI descritta in precedenza.
- *sessione di testing*: è la fase di acquisizione *on-line*. Il segnale EEG acquisito dal soggetto, viene elaborato, ed in *real time* viene fornito l'output del sistema in base

al risultato della classificazione (*real feedback*). Il numero di blocchi random varia da 4 (caso migliore) a 23 (caso peggiore). In termini di stimoli si ha un minimo di 13 stimoli ad un massimo di 92. Se nel caso di learning la sessione si completa sempre con il raggiungimento del target, in una sessione di testing dipende da quanti veri positivi (eventi target riconosciuti come tali), falsi positivi (eventi standard classificati come target) e falsi negativi (eventi target classificati come standard) vengono riconosciuti dal classificatore. Infatti, il feedback generato produce lo spostamento del cursore verso la direzione della freccia rispetto cui è stata acquisita la traccia da elaborare. Se è un vero positivo, allora il cursore si muove correttamente verso il target; se è un falso positivo, allora il cursore si muove in una direzione che non è quella del target; se si ha un falso negativo, il cursore non si muove e quello che succede è che il blocco random che serve per elicitare la P300 non è stato “sfruttato” e quindi sarà necessario prolungare la prova. Per via di questi errori, si aumentano il numero di tentativi a disposizione fino ad un massimo di 92, limite imposto affinché la prova abbia una durata finita (con 92 stimoli la durata è di circa 4 minuti). Da notare che una sessione può concludersi in tre diverse condizioni:

1. fine sessione per raggiungimento del target;
2. fine sessione per numero massimo di stimoli senza il raggiungimento del target;
3. fine sessione per raggiungimento di uno dei tre eventi standard.

La modalità con cui viene condotto il protocollo sperimentale è indicato in figura 2.3.1, in cui *Acquisition* e *Testing* fanno riferimento rispettivamente alle sessioni di learning e testing precedentemente spiegate.

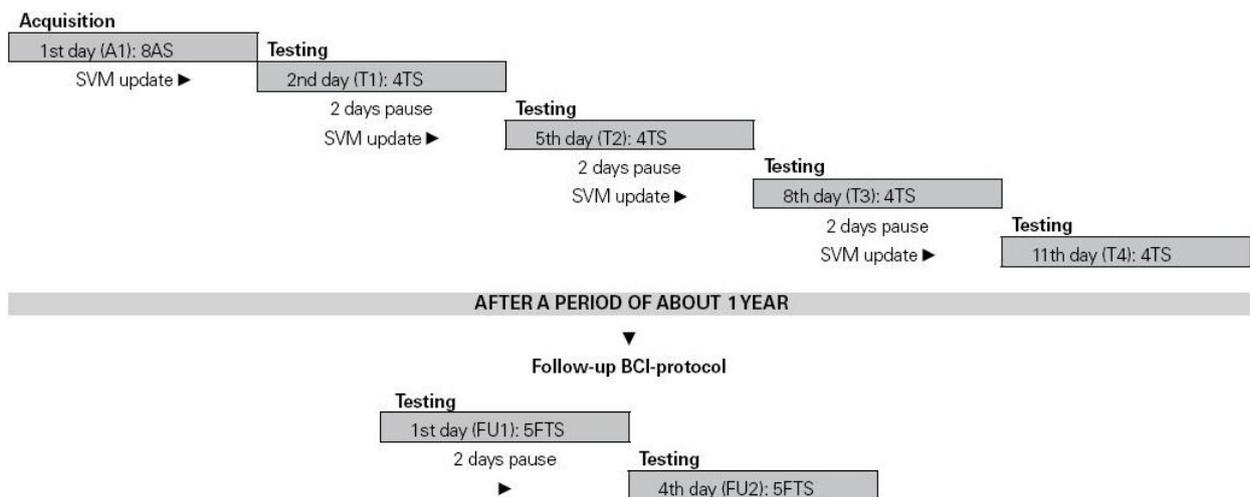


Figura 2.3.1: *Protocollo Sperimentale svolto nell'istituto IRCCS San Camillo [39].*

Il primo giorno si applica la sessione di *learning*, la quale serve, come detto, ad acquisire i dati dal soggetto per preparare il primo classificatore. Da qui in poi, la tipologia di

sessioni che verrà applicata è di *testing* ed in questa il soggetto dovrà eseguire quattro sessioni ciascuna con un target.

In particolare, nel primo giorno di testing verrà usato il classificatore addestrato sui dati del primo giorno di learning; nel secondo giorno di testing verrà utilizzato il classificatore addestrato sulla base dei dati acquisiti nella sessione di testing precedente e così via. Nell'ultimo giorno di testing non verrà più aggiornato il classificatore e quindi i dati raccolti servono unicamente ad indagini statistiche per giudicare la performance del sistema usato. Sorge spontaneamente una questione: perché fare più sessioni di testing? In linea teorica, già con la prima sessione di testing si potevano ricavare delle informazioni sulle prestazioni. In realtà non è così ed una semplice definizione ci permette di capire ciò: *il sistema BCI è un'abilità* [43]. Si potrebbe considerare un solo giorno di testing esclusivamente se il soggetto mostra fin da subito delle performance molto stabili. Fattori come la motivazione, la qualità di vita, la depressione e l'umore possono giocare un ruolo fondamentale nei processi cognitivi traducibili quindi in migliori o peggiori performance del soggetto. Oppure la presenza di stimoli visivi in grado di evocare nella mente del soggetto particolari emozioni. In [26] è condotta un'interessante analisi proprio in questo senso, in cui persino l'ansia da prestazione durante l'esperimento gioca un ruolo fondamentale. Inoltre, si parla di *motivazione estrinseca* [26] per i soggetti sani, i quali si sottopongono agli esperimenti per una propria ragione o semplicemente per un compenso monetario; si parla invece di *motivazione intrinseca* [26] per i pazienti ALS, i quali vedono in un possibile futuro la BCI come uno strumento quotidianamente utilizzabile nelle loro vite. Questo spiegherebbe come spesso nei protocolli sperimentali, le prestazioni dei soggetti malati sono significativamente superiori a quelli dei soggetti sani.

Detto questo, affinché si possa rendere più stabile il classificatore è necessario eseguire un aggiustamento periodico dei suoi valori e questo si realizza durante le tre giornate di testing.

In figura è anche mostrato il protocollo di follow up eseguito da alcuni soggetti dopo un anno. Proprio perché la BCI è un'abilità, si vuole verificare se tale rimane anche nel tempo e senza un uso prolungato del sistema.

2.4 Conclusioni

In questo capitolo è stato descritto il sistema BCI dell'IRCCS San Camillo basato sul paradigma oddball con stimolazione di tipo visivo e che ha generato i dati che verranno utilizzati per la validazione off-line delle tecniche bayesiane di stima. Tali metodi verranno presentati negli aspetti tecnici nei prossimi capitolo 3 e 4, prima di passare all'analisi dei risultati nei capitoli 5 e 6.

Capitolo 3

Un metodo bayesiano di stima single-trial a due passi (B2S)

3.1 Approccio generale alla stima bayesiana a minima varianza d'errore

In generale, stimare il valore della variabile incognita $x \in \mathcal{R}^p$ sulla base delle osservazioni y della variabile aleatoria $Y \in \mathcal{R}^n$ è un problema che può essere affrontato in due possibili scenari:

1. Stima parametrica: la variabile x è un parametro incognito certo (deterministico) e la densità di probabilità di Y dipende unicamente da $X \rightarrow f_Y^X(y)$.
2. Stima bayesiana: la variabile incognita X è una variabile aleatoria, di cui si vuole quindi stimare una sua realizzazione x , ed è nota la densità di probabilità congiunta di Y e $X \rightarrow f_{X,Y}(x, y)$.

In questo capitolo, la stima d'interesse è quella bayesiana, ambito in cui non solo si dispone delle conoscenze sul legame fra grandezza da stimare x e la misura y , ma anche di informazioni a priori di tipo probabilistico sull'incognita, come ad esempio

- la densità di probabilità a priori del vettore $X \rightarrow p_X(x)$;
- la densità di probabilità di Y condizionata da $X \rightarrow f_{y/x}(y/x)$, chiamata *likelihood function*.

L'ambito è quello dell'inferenza statistica, che consiste nel ricavare informazioni sul vettore aleatorio incognito X , a partire dalle osservazioni del vettore aleatorio y . In particolare, ciò che si sta cercando è la densità di probabilità di x condizionata all'evento di osservazione $Y=y$. Questa relazione è fornita dalla formula di Bayes:

$$f_{x|y}(x|Y=y) = \frac{f_{y|x}(y|x) p_x(x)}{\int_{\mathcal{R}^n} f_{y|x}(y|x) p_x(x) dx} \quad (3.1.1)$$

L'obiettivo è quindi, data l'osservazione della realizzazione y del vettore aleatorio Y , determinare, con la migliore approssimazione possibile, quale è il valore campionario di x del vettore aleatorio X , non osservabile, che ha contribuito in qualche modo a produrla. Per risolvere la 3.1.1 è necessario impostare una funzione costo. In questo caso specifico si sceglierà quel valore di x che minimizza la seguente espressione $E[||x - \hat{x}||^2]$, ovvero il valore atteso della norma al quadrato dell'errore di stima.

La stima desiderata, detta stima a minimo errore quadratico medio, coincide con il valore $E[X|Y = y]$. L'entità $E[X|Y = y]$ è definita *media condizionata* ed anche *valor medio a posteriori*.

L'espressione relativa è la seguente:

$$\hat{x} = E[X|Y = y] = \int_{R^n} x f_{x|y}(x|Y = y) dx = \int_{R^n} x \frac{f_{y|x}(y|x) p_x(x)}{\int_{R^n} f_{y|x}(y|x) p_x(x) dx} dx \quad (3.1.2)$$

Per ottenere la formulazione analitica dell'espressione 3.1.2 bisognerà introdurre delle ipotesi vincolanti al problema stesso.

Nel prossimo paragrafo si descrivono così tali assunzioni per la stima B2S.

3.2 Assunzioni del metodo bayesiano a due passi

Il metodo B2S [37, 38] è un metodo di stima single-trial di potenziali evento-correlati il cui fine è quello di ricostruire la componente ERP a partire dal segnale grezzo fornito. Con il termine *single trial* si specifica che l'output dell'algoritmo è la stima della singola traccia, ottenuta attraverso un processo di "estrazione" dal resto del segnale EEG, interpretato così come rumore.

Il modello dei dati adottato per la stima bayesiana è il seguente

$$y_j(t) = u_j(t) + v_j(t) \quad (3.2.1)$$

in cui $y_j(t)$ è il segnale registrato per la sweep j -esima, $v_j(t)$ è il segnale EEG interpretato come rumore di fondo ed infine $u_j(t)$ è la componente ERP da stimare. Assumendo che $u_j(t)$ e $v_j(t)$ abbiano natura aleatoria, chiaramente anche $y_j(t)$ può essere interpretato come una grandezza stocastica.

Per semplicità, ipotizziamo che la stima \hat{u}_j sia legata da una qualche funzione al vettore delle misure $y_j(t)$ e che, in particolare, tale legame sia di tipo lineare:

$$\hat{u}_j = f(y_j) = K * y_j + m$$

dove K è lo STIMATORE LINEARE cercato. Sotto queste ipotesi, la miglior stima di $u_j(t)$ dato $y_j(t)$ è fornita dalla seguente espressione analitica:

$$\hat{u}_j = E[u_j|y_j] = \mu + \left(\Sigma_{v_j}^{-1} + \Sigma_{u_j}^{-1}\right)^{-1} \Sigma_{v_j}^{-1} (y_j - \mu) \quad (3.2.2)$$

Da notare che non è stata introdotta nessuna ipotesi sulla media dei segnali $y_j(t)$ e $u_j(t)$ e questo ne ha determinato nella formula 3.2.2 la presenza del vettore μ , che rappresenta

il valore atteso a priori del segnale incognito. Infatti, visto che il segnale $u_j(t)$ rappresenta la componente ERP, il segnale μ può essere interpretato come il profilo atteso della componente P300. Questo riferimento verrà calcolato nel primo passo del metodo, mentre nel secondo esso sarà utilizzato nella 3.2.2 per fornire le stime single-trial finali.

Affinché la 3.2.2 sia implementabile nell'algoritmo, è necessario definire i parametri in essa presenti. Oltre il vettore μ , bisogna valutare adesso le due matrici

1. $\Sigma_v \rightarrow$ matrice di covarianza dell'errore di misura;
2. $\Sigma_u \rightarrow$ matrice di covarianza del segnale $u_j(t)$.

3.2.1 Descrizione dell'errore di misura

Si ricorda nuovamente che per errore di misura si intende anche tutta la parte di EEG non coinvolta nel complesso ERP e rappresentata dalla maggior parte dell'attività a riposo. Tale componente è così espressa dal segnale $v_j(t)$, della quale è possibile fornire una descrizione quantitativa, ovvero i valori numerici della media e della varianza.

Il vettore $v_j(t)$ si suppone a media nulla e correlato, descrivibile come un processo autoregressivo (AR). Quest'ultima ipotesi rende la matrice Σ_{v_j} non diagonale ed in particolare, caratterizzata dalla seguente espressione

$$\Sigma_{v_j} = \sigma_j^2 * (A_j^T A_j)^{-1} \quad (3.2.3)$$

A_j è una matrice di Toeplitz triangolare inferiore, con rango pari alla lunghezza del vettore $v_j(t)$ e la prima colonna pari al vettore $[1 \ a_{1j} \ a_{2j} \ \dots \ a_{pj} \ \dots \ 0 \ 0 \ \dots \ 0]^T$. I coefficienti $a_{1j} \ \dots \ a_{pj}$ sono i coefficienti del modello AR identificato rispetto al pre-stimolo della sweep corrente $y_j(t)$, p è l'ordine del modello e σ_j^2 è la varianza del rumore bianco che pilota il filtro. La scelta dell'ordine ottimo è stata fatta mediante il criterio di Akaike Information Criterion (AIC), facendo variare il range di valori da 2 a 14.

Il modello AR viene identificato rispetto al pre-stimolo del segnale poiché si assume che in questa parte non vi sia potenziale evento-correlato. Si assume quindi che il comportamento del rumore EEG prima dello stimolo possa essere esteso anche alla parte successiva di post-stimolo e che questa informazione possa essere utilizzare per estrarre l'ERP dal rumore in cui è sommerso. Condizione necessaria per l'identificazione del modello AR è che il segnale sia stazionario. Poiché il segnale EEG non può essere considerato stazionario al di sopra dei due secondi, si considera adatta per l'identificazione del modello la parte di prestimolo pari a 500 ms.

L'ultima osservazione è relativa alla presenza del pedice j nella matrice Σ_v . Questo significa che la matrice di covarianza dell'errore di misura dipende dalla sweep corrente. Infatti, ogni matrice verrà identificata rispetto al pre-stimolo proprio della sweep. Quindi, ogni sweep avrà la relativa matrice Σ_{v_j} .

3.2.2 Descrizione dell'informazione a priori del segnale incognito

$$u_j(t)$$

Nel paragrafo 3.1 è stato messo in evidenza come la stima bayesiana abbia una componente importante, rappresentata dall'informazione a priori che si fornisce sul segnale incognito. L'ipotesi che viene fatta in questo caso riguarda la regolarità del segnale incognito $u_j(t)$. In particolare, si assume che il segnale possa essere rappresentato mediante un modello *random walk*, semplice od integrato.

Si propone l'equazione del modello semplice

$$u_k = u_{k-1} + w_k \quad k = 1, 2, \dots, n \quad (3.2.4)$$

L'idea alla base di tale modello è che se il processo aleatorio è regolare, allora è molto improbabile che la differenza fra due campioni adiacenti subisca variazioni brusche. E' invece più verosimile che un campione ed il suo precedente assumano valori prossimi fra loro, tale per cui si può definire "piccola" la distanza fra gli stessi. In questo modo, si ipotizza l'esistenza di un modello stocastico (quello nella 3.2.4) tale per cui la differenza $\Delta u_k = u_k - u_{k-1}$ sia una variabile aleatoria w_k , della quale si può unicamente fornire una descrizione qualitativa: media nulla, varianza finita *incognita* λ^2 e campioni incorrelati. Il fatto che la varianza λ^2 sia incognita, suggerisce che questo parametro dovrà essere stimato in qualche modo all'interno dell'algoritmo. Il come si vedrà successivamente.

Infine, si ipotizza che il segnale incognito $u_j(t)$ sia causale, ovvero che $u_k = 0$ per $k \leq 0$. L'obiettivo che bisogna raggiungere è la determinazione della matrice Σ_{u_j} , contenente tutta l'informazione a priori per il segnale ERP contenuto nella sweep j -esima. Questa verrà stimata proprio dalla matrice di covarianza della variabile w_k che è pari a $\Sigma_w = \lambda^2 \mathbf{I}$, essendo w_k a campioni scorrelati e con varianza λ^2 . L'equazione 3.2.4 può essere generalizzata in forma matriciale come

$$Fu = w \quad (3.2.5)$$

con $F = \Delta^m$ matrice delle differenze m -esime. Nel caso particolare indicato nella 3.2.4 $m = 1$ ed il segnale è descritto da un modello random walk semplice. Dalla 3.2.5, la matrice Σ_{u_j} ha la seguente espressione

$$\Sigma_{u_j} = \lambda_j^2 (F_j^T F_j)^{-1} \quad (3.2.6)$$

in cui si ricorda che il parametro λ_j^2 è incognito.

Come sarà chiaro in seguito, lo stesso modello sarà utilizzato nel secondo passo del metodo per modellare lo scostamento del potenziale evento-correlato single-trial dal potenziale medio stimato al primo passo. Il modello 3.2.4 sarà modificato nel seguente modo:

$$u_k - \mu_k = u_{k-1} - \mu_{k-1} + w_k \quad (3.2.7)$$

in cui il vettore μ rappresenta il vettore di riferimento, cioè il profilo atteso del ERP. Questo modello adesso non fa più ipotesi sulla distanza fra un campioni ed il precedente

del segnale incognito, bensì ipotizza che la distanza fra il campione del segnale ed il vettore di riferimento all'istante k -esimo e quella fra il campione del segnale incognito e del riferimento all'istante $k-1$, siano valori prossimi tra loro. Quindi, non si effettua più il confronto diretto fra i valori del segnale incognito, ma rispetto a quanto questi si discostano dal vettore di riferimento μ . La formula 3.2.6 per la Σ_{u_j} è ancora valida proprio perché dipende unicamente dal valore incognito λ_j^2 e dalla matrice F_j . Tra l'altro, il vettore μ è il medesimo che compare nella 3.2.2.

3.2.3 Calcolo del vettore di riferimento μ

Il calcolo del vettore di riferimento μ costituisce il primo passo del metodo B2S. Il valore ottenuto sarà poi utilizzato nel secondo passo del metodo che consiste nella stima vera e propria degli ERPs single-trial secondo la formula 3.2.2. Il riferimento viene stimato attraverso la seguente procedura

1. Applicazione dello stimatore 3.2.2 ponendo $\mu = 0$

Ponendo μ uguale a zero, la formula 3.2.2 si riscrive come di seguito

$$\hat{u}'_j = E[u'_j | y_j] = \left(\Sigma_{v_j}^{-1} + \Sigma_{u'_j}^{-1} \right)^{-1} \Sigma_{v_j}^{-1} y_j \quad (3.2.8)$$

Le stime in output saranno così le $\hat{u}'_j(t)$.

2. Calcolo finale del vettore di riferimento μ

Il riferimento μ viene calcolato come media pesata delle stime $\hat{u}'_j(t)$

$$\mu = \frac{\sum_{i=1}^N c_i \hat{u}'_i}{\sum_{i=1}^N c_i} \quad (3.2.9)$$

in cui i pesi c_i sono inversamente proporzionali al valore atteso della norma dell'errore di stima. Questa quantità è uguale alla traccia della matrice di covarianza dell'errore di stima $cov(\tilde{u}_j) = \sigma_j^2 (A_j^T A_j + \gamma_j F_j^T F_j)^{-1}$, dove \tilde{u}_j è appunto l'errore di stima. Il significato del parametro γ_j verrà indicato nel prossimo paragrafo.

3.3 Formulazione finale e parametro di regolarizzazione

In questa sezione si sostituiranno le relazioni per la Σ_{v_j} e Σ_{u_j} nella formulazione 3.2.2, ottenendo:

$$\hat{u}_j = \mu + (A_j^T A_j + \gamma_j F_j^T F_j)^{-1} A_j^T A_j (y_j - \mu) \quad (3.3.1)$$

Mentre, la 3.2.8 per il riferimento μ può essere riscritta come:

$$\hat{u}'_j = \left(A_j'^T A_j' + \gamma_j' F_j'^T F_j' \right)^{-1} A_j'^T A_j' y_j \quad (3.3.2)$$

E' da notare che la matrice A_j della formula 3.3.1 è diversa rispetto a quella impiegata nel calcolo del riferimento della formula 3.3.2: infatti nel caso del riferimento, il pre-stimolo rispetto cui viene identificato il modello AR è quello relativo alla sweep grezza; mentre, nel caso della formula finale, il pre-stimolo utilizzato per i modelli AR è quello relativo alla sweep $y_j^* = y_j - \mu$.

Rimane così da definire il parametro γ_j .

Esso è il parametro di regolarizzazione, definito dalla seguente espressione

$$\gamma_j = \frac{\sigma_j^2}{\lambda_j^2} \quad (3.3.3)$$

in cui σ_j^2 è la varianza del rumore d'ingresso al modello AR trovato per la sweep j-esima (quindi il suo valore è noto), mentre λ_j^2 è la varianza incognita introdotta con il modello random-walk.

Il parametro γ_j è appunto detto di regolarizzazione proprio perché rappresenta una sorta di peso fra uno stimatore che privilegia maggiormente l'aderenza alle misure (caso limite: $\gamma_j \rightarrow 0$, lo stimatore non sfrutta l'informazione a priori generando il fenomeno di *overfitting*) e quello che invece pesa maggiormente l'informazione a priori del segnale incognito (caso limite: $\gamma_j \rightarrow \infty$, lo stimatore non si cura dell'aderenza ai dati generando il fenomeno di *oversmoothing*).

Il calcolo del parametro γ_j viene effettuato utilizzando il criterio di Discrepanza (Twomey), basato sulla definizione di una variabile che rappresenta la somma pesata dei residui al quadrato, definita dalla seguente variabile aleatoria:

$$WRSS(\gamma_j) = r(\gamma_j)^T * A_j^T A_j r(\gamma_j)$$

dove $r(\gamma_j) = y_j - \hat{u}_j$ è il vettore dei residui, il quale, essendo in funzione di \hat{u}_j , risulta dipendere da γ_j . Il criterio di discrepanza trova il valore ottimo di γ_j tale da soddisfare la seguente espressione:

$$WRSS(\gamma_j) = n\sigma_j^2$$

Chiaramente, questo criterio verrà applicato una volta per il calcolo del riferimento, e la seconda volta per la stima finale del segnale $u_j(t)$, quindi le formule soprascritte vengono tutte particolarizzate con le rispettive variabili.

3.4 Conclusioni

In questo capitolo si è introdotto il metodo bayesiano a due passi B2S, una tecnica single trial che riceve in input le sweeps grezze rispetto le quali voler stimare la componente P300 e fornisce in uscita una stima della componente ERP. E' un metodo su due step, il primo riferito al calcolo del riferimento μ e il secondo riferito alla stima finale delle sweeps $u_j(t)$. Per quanto concerne il vettore μ , lo stimatore utilizzato è il medesimo della formulazione finale, sotto ipotesi che il vettore di riferimento sia nullo. L'informazione così ottenuta viene utilizzata per il calcolo delle stima finali $u_j(t)$ secondo quanto indicato nella 3.3.2.

Nel prossimo capitolo verrà introdotto il metodo di Multi Task Learning, per poi passare all'analisi delle stime dai dati simulati nel capitolo 5 e dai dati reali nel capitolo 6.

Capitolo 4

Un metodo di Multi Task Learning per la stima single-trial (MTL)

4.1 Introduzione al metodo MTL

Il secondo metodo di stima, descritto nel presente capitolo, utilizza ancora un approccio bayesiano. In particolare, le stime dell'ERP medio e degli ERPs single-trial avvengono in un solo passo, utilizzando tutte le sweeps a disposizione all'interno del dataset. Il metodo sarà indicato con l'acronimo MTL (Multi Task Learning) che indica il fatto che l'informazione disponibile per una sweep viene usata per la stima di tutte le altre secondo un approccio noto anche come analisi di popolazione [29].

In questo caso, la popolazione omogenea è l'insieme delle sweeps acquisite da uno stesso soggetto. Similmente al metodo B2S, si opera ancora in un contesto bayesiano.

4.2 Assunzioni del Multi Task Learning

A partire da una popolazione omogenea di N sweeps, la generica traccia grezza registrata $y_j(t)$ può essere pensata come la somma di due componenti

$$y_j(t) = z_j(t) + v_j(t) \quad j = 1, 2, \dots, N \quad (4.2.1)$$

dove $z_j(t)$ è l'ERP incognito, $v_j(t)$ è il rumore ovvero l'EEG di fondo che si assume non coinvolto nella generazione del potenziale ERP.

In particolare, si assume che $z_j(t)$ possa essere rappresentata dal seguente modello

$$z_j(t) = \bar{z}(t) + \tilde{z}_j(t) \quad (4.2.2)$$

in cui $\bar{z}(t)$ è la curva media, la quale indica delle proprietà “medie” della popolazione; $\tilde{z}_j(t)$ è lo shift della singola curva j rispetto alla componente media $\bar{z}(t)$. Da notare che la curva $\bar{z}(t)$ non è dipendente dal pedice j proprio perché è una componente comune a tutte le tracce della medesima popolazione. E' possibile pensare nel caso in cui la $\bar{z}(t)$ sia nulla che tutte le sweeps appartenenti alla popolazione omogenea siano tra loro indipendenti. Proprio perché la stima viene fatta utilizzando tutte le sweeps a disposizione, è necessario

introdurre una notazione matriciale in grado di fornire una scrittura compatta

$$\begin{aligned}\mathbf{y} &= [y_1(t_1) \dots y_1(t_m); \dots; y_N(t_1) \dots y_N(t_m)]^T \\ \bar{\mathbf{z}} &= [\bar{z}(t_1) \dots \bar{z}(t_m); \dots; \bar{z}(t_1) \dots \bar{z}(t_m)]^T \\ \tilde{\mathbf{z}} &= [\tilde{z}_1(t_1) \dots \tilde{z}_1(t_m); \dots; \tilde{z}_N(t_1) \dots \tilde{z}_N(t_m)]^T \\ \mathbf{v} &= [v_1(t_1) \dots v_1(t_m); \dots; v_N(t_1) \dots v_N(t_m)]^T\end{aligned}$$

in cui m è il numero totale di campioni per ogni sweep e nel caso in esame è pari a 300. In forma matriciale, la formula 4.2.1, considerando l'espressione di z_j nella 4.2.2, si può riscrivere come segue

$$\mathbf{y} = \bar{\mathbf{z}} + \tilde{\mathbf{z}} + \mathbf{v} \quad (4.2.3)$$

L'approccio di stima adottato nel Multi Task Learning è ancora quello bayesiano (vedi sezione 3.1), per cui ciò che adesso bisogna introdurre sono le assunzioni per esplicitare la formula 3.1.2. In particolare si assume che le variabili $\bar{\mathbf{z}}$, $\tilde{\mathbf{z}}$ e \mathbf{v} abbiano natura aleatoria, e che siano, inoltre, processi indipendenti e CONGIUNTAMENTE GAUSSIANI. Quest'ultima è un'ipotesi molto forte tale da permettere di esplicitare la 3.1.2 ottenendo un'espressione analitica, in forma generale, che rappresenta la miglior stima in assoluto di x dato $Y = y$

$$\hat{x} = E[x|Y = y] = \mu_x + \Sigma_{xy}\Sigma_y^{-1}(\mathbf{y} - \mu_y) \quad (4.2.4)$$

Nel caso in esame, sia la curva $\bar{z}(t)$ che gli shift individuali $\tilde{z}_j(t)$ verranno stimate dall'equazioni 4.2.4, in cui la variabile x verrà sostituita da $\bar{z}(t)$ e $\tilde{z}_j(t)$, stime condotte chiaramente in modo separato.

Prima di passare alla formulazione esplicita per le due stime, è necessario definire

1. Matrice di covarianza dell'errore di misura $\rightarrow \Sigma_v$

Il rumore di misura è assunto a media nulla, gaussiano e di natura correlata, per cui la sua identificazione è uguale a quanto già introdotto nel metodo B2S, quindi si farà, anche in questo caso, riferimento a quanto riportato nella sezione 3.2.1.

2. Valori attesi dei vettori aleatori in gioco

- a) Valore atteso di $\bar{z}(t)$

$$E[\bar{z}(t)] = 0 \quad (4.2.5)$$

Questa assunzione è facilmente verificata sottraendo la media ai dati.

- b) Valore atteso di $\tilde{z}_j(t)$

è ragionevolmente posto a zero proprio perché rappresenta lo shift rispetto alla curva media

$$E[\tilde{z}_j(t)] = 0 \quad (4.2.6)$$

- c) Valore atteso di $y_j(t)$

In riferimento all'espressione 4.2.3, la 4.2.5 e 4.2.6 e ricordando che $E[v] = 0$ perché il rumore è ipotizzato a media nulla, si ottiene

$$E[y_j] = 0 \quad (4.2.7)$$

4.3 Stima della curva media $\bar{z}(t)$

Si considera 4.2.4. In termini di $\bar{z}(t)$ si ottiene

$$\hat{\bar{z}}(t) = E[\bar{z}(t)|\mathbf{y}] = E[\bar{z}(t)] + cov[\bar{z}(t), \mathbf{y}] * var[\mathbf{y}]^{-1} * (\mathbf{y} - E[\mathbf{y}]) \quad (4.3.1)$$

Inoltre, considerando la 4.2.5 e la 4.2.7, la precedente espressione può essere riscritta come segue

$$\hat{\bar{z}}(t) = E[\bar{z}(t)|y] = cov[\bar{z}(t), y] * var[y]^{-1} * y \quad (4.3.2)$$

Affinché sia possibile determinare la 4.3.2, bisogna esplicitare i due termini principali: $cov[\bar{z}(t), y]$ e $var[\mathbf{y}]^{-1} y$.

1. $cov[\bar{z}(t), y]$

tenendo presente che i vettori \bar{z} , \tilde{z} e v sono indipendenti fra loro e facendo riferimento alla 4.2.3, si può scrivere:

$$\begin{aligned} cov[\bar{z}(t), \mathbf{y}] &= cov[\bar{z}(t), \bar{\mathbf{z}} + \tilde{\mathbf{z}} + \mathbf{v}] = cov[\bar{z}(t), \bar{\mathbf{z}}] = \\ &= cov[\bar{z}(t), [z(t_1) \dots z(t_m)]] = \\ &= [\bar{R}(t, t_1) \dots \bar{R}(t, t_m)] \end{aligned} \quad (4.3.3)$$

2. $var[\mathbf{y}]^{-1} y$

A partire dal modello 4.2.3, si può scrivere la seguente espressione

$$var[\mathbf{y}] = var[\bar{\mathbf{z}}] + var[\tilde{\mathbf{z}}] + \Sigma_v \quad (4.3.4)$$

con

$$var[\bar{\mathbf{z}}] = \bar{\mathbf{R}} := \begin{bmatrix} \bar{R}(t_1, t_1) & \cdots & \bar{R}(t_1, t_m) \\ \vdots & \ddots & \vdots \\ \bar{R}(t_m, t_1) & \cdots & \bar{R}(t_m, t_m) \end{bmatrix}$$

$$var[\tilde{\mathbf{z}}] = \tilde{\mathbf{R}} := blockdiag\{\tilde{\mathbf{R}}_1, \dots, \tilde{\mathbf{R}}_N\}$$

$$\tilde{\mathbf{R}}_j = \begin{bmatrix} \tilde{R}_j(t_1, t_m) & \cdots & \tilde{R}_j(t_1, t_m) \\ \vdots & \ddots & \vdots \\ \tilde{R}_j(t_1, t_m) & \cdots & \tilde{R}_j(t_1, t_m) \end{bmatrix}$$

Sostituendo le espressioni trovate nella 4.3.2 e definendo il vettore \mathbf{c} come segue

$$\mathbf{c} = \Sigma_y^{-1} \mathbf{y} = [c_{1_1} \dots c_{1_m}; \dots; c_{N_m} \dots c_{N_m}]^T \quad (4.3.5)$$

si ottiene la seguente espressione per la stima

$$\hat{\bar{z}}(t) = E[\bar{z}(t) | \mathbf{y}] = [\bar{R}(t, t_1^1) \dots \bar{R}(t, t_{mN}^N)] [c_1^1 \dots c_{n_1}^1; \dots; c_1^N \dots c_{n_N}^N]^T = \sum_{j=1}^N \sum_{k=1}^{n_j} c_k^j \bar{R}(t, t_k^j) \quad (4.3.6)$$

4.4 Stima delle curve individuali $z_j(t)$

La stima delle curve individuali è analoga a quanto visto per la stima della curva di popolazione. In particolare, in riferimento alla 4.2.2, stimare la $z_j(t)$ significa stimare singolarmente la $\bar{z}_j(t)$ e la $\tilde{z}_j(t)$, dato che sono stati considerati due processi indipendenti. L'espressione della $\bar{z}_j(t)$ è stata ricavata nella 4.3.6. Bisogna adesso calcolare la $\tilde{z}_j(t)$. La formula 4.2.4, viene così riscritta nel seguente modo

$$\hat{\tilde{z}}_j(t) = E[\tilde{z}_j(t) | \mathbf{y}] = E[\tilde{z}_j(t)] + cov[\tilde{z}_j(t), \mathbf{y}] * var[\mathbf{y}]^{-1} * (\mathbf{y} - E[\mathbf{y}]) \quad (4.4.1)$$

Questa espressione si semplifica in funzione della 4.2.6 e 4.2.7 nel seguente modo

$$\hat{\tilde{z}}_j(t) = cov[\tilde{z}_j(t), \mathbf{y}] * var[\mathbf{y}]^{-1} * \mathbf{y} \quad (4.4.2)$$

Il termine $var[\mathbf{y}]^{-1} * \mathbf{y}$ è noto e viene calcolato come descritto nella 4.3.5; bisogna così esplicitare il termine $cov[\tilde{z}_j(t), \mathbf{y}]$
 $cov[\tilde{z}_j(t) | \mathbf{y}] = cov[\tilde{z}_j(t), \bar{\mathbf{z}} + \tilde{\mathbf{z}} + \mathbf{v}]$, ma poiché i vettori $\bar{\mathbf{z}}, \tilde{\mathbf{z}}$ e \mathbf{v} sono indipendenti e quindi la mutua covarianza è nulla, si può riscrivere

$$cov[\tilde{z}_j(t) | \mathbf{y}] = cov[\tilde{z}_j(t), \tilde{\mathbf{z}}] = cov[\tilde{z}_j(t), \tilde{\mathbf{z}}_1, \dots, \tilde{\mathbf{z}}_N] = [\tilde{R}_j(t, t_1) \dots \tilde{R}_j(t, t_m)] \quad (4.4.3)$$

In questo modo, sostituendo la 4.4.3 e la 4.3.5 nella 4.4.1, si ottiene la seguente espressione per la stima

$$\hat{\tilde{z}}_j(t) = [\tilde{R}_j(t, t_1) \dots \tilde{R}_j(t, t_m)] [c_{1_1} \dots c_{1_m}; \dots; c_{N_1} \dots c_{N_m}]^T = \sum_{k=1}^{n_j} c_k^j \tilde{R}(t, t_k) \quad (4.4.4)$$

Infine, si ricava la $z_j(t)$, stima della curva individuale del soggetto j , come segue

$$\begin{aligned} \hat{z}_j &= E[z^j(t) | \mathbf{y}] = E[\bar{z}(t) + \tilde{z}_j(t) | \mathbf{y}] = E[\bar{z}(t) | \mathbf{y}] + E[\tilde{z}_j(t) | \mathbf{y}] \\ &= \hat{\bar{z}}(t) + \hat{\tilde{z}}_j(t) \end{aligned} \quad (4.4.5)$$

in cui $\hat{\bar{z}}(t)$ è indicata nella 4.3.6 e la $\hat{\tilde{z}}_j(t)$ nella 4.4.4.

4.5 Modelli di Wiener e determinazione degli iper-parametri

Nelle precedenti sezioni, si sono trovate le forme esplicite per la stima della curva media di popolazione e delle curve individuali, rispettivamente indicate dalla 4.3.6 e 4.4.5. In entrambe le espressioni però, la funzione di auto-covarianza non è nota. E' necessario così introdurre dei modelli che riflettano le conoscenze a priori a disposizione sui segnali incogniti. Si assume che i segnali da stimare siano "regolari" e si utilizzano dei processi di Wiener integrati, per modellare sia la curva media che gli shift individuali. Sotto questa ipotesi è possibile ottenere le seguenti espressioni per la matrice di covarianza dei processi $\bar{\mathbf{z}}$ e $\tilde{\mathbf{z}}$

$$\bar{R}(t, \tau) = \bar{\lambda}^2 \begin{cases} \frac{t^2}{2} (\tau - \frac{t}{3}) & t \leq \tau \\ \frac{\tau^2}{2} (t - \frac{\tau}{3}) & t > \tau \end{cases} \quad (4.5.1)$$

$$\tilde{R}(t, \tau) = \tilde{\lambda}^2 \begin{cases} \frac{t^2}{2} (\tau - \frac{t}{3}) & t \leq \tau \\ \frac{\tau^2}{2} (t - \frac{\tau}{3}) & t > \tau \end{cases} \quad (4.5.2)$$

dove $\bar{\lambda}^2$ e $\tilde{\lambda}^2$ sono le varianze dei rumori bianchi gaussiani a media nulla in ingresso ai modelli di Wiener che descrivono i due segnali. Questi sono parametri sono incogniti e detti *iper-parametri*.

Il metodo di stima è il cosiddetto *bayesiano empirico*: si basa sull'applicazione in primis della stima Maximum Likelihood per il calcolo di $(\bar{\lambda}^2, \tilde{\lambda}^2)$ e in secundis sull'utilizzo degli stessi nella stima bayesiana. I parametri $(\bar{\lambda}^2, \tilde{\lambda}^2)$ vengono inseriti nel vettore $\Psi = [\bar{\lambda}^2, \tilde{\lambda}^2]$ e si cercherà quel valore di Ψ che massimizza la funzione di verosimiglianza. Si dimostra che i valori degli iper-parametri cercati minimizzano la seguente espressione

$$\Psi = \underset{\Psi}{\operatorname{argmin}} \{ \ln(\det(\operatorname{var}[\mathbf{y}])) + \mathbf{y}^T * \Sigma_{\mathbf{y}}^{-1} * \mathbf{y} \} \quad (4.5.3)$$

Calcolati gli iper-parametri e sostituiti nelle relative formule, si ottengono così la curva media $\bar{z}(t)$ e gli shift individuali $\tilde{z}_j(t)$, nonché le curve individuali stimate $z_j(t)$.

4.6 Conclusioni

In questo capitolo si è descritta la parte teorica alla base delle tecnica Multi Task Learning. Anche questo metodo, come il B2S, utilizza un approccio bayesiano. In particolare, l'ERP medio e gli ERPs single-trial vengono stimati, in un solo passo, in un contesto di popolazione attraverso l'analisi congiunta di tutte le sweeps a disposizione.

Il capitolo 3 e il capitolo corrente hanno così introdotto le tecniche da un punto di vista

4. UN METODO DI MULTI TASK LEARNING PER LA STIMA SINGLE-TRIAL (MTL)

teorico. I capitoli successivi sono dedicati all'analisi delle stime fornite dai due metodi, rispettivamente per i dati simulati nel capitolo 5 e per i dati reali nel capitolo 6.

Capitolo 5

Applicazione delle tecniche B2S e MTL per la stima della P300 da dati simulati

Questo capitolo è suddiviso in tre sezioni principali. La prima illustra il processo di simulazione dei dati, sia per le sweeps non target che per le sweeps target. Nella seconda e terza vengono invece mostrati i risultati delle stime della componente P300 dai metodi B2S e MTL.

5.1 Generazione dei dati simulati

5.1.1 Simulazione delle sweeps non target

Le sweeps non target sono caratterizzate dall'assenza della componente P300. Si può assumere che esse siano delle tracce di EEG in cui non si è verificato alcun evento rilevante. In termini di simulazione, basterà quindi conoscere il solo rumore di fondo, il quale viene generato come descritto in seguito.

5.1.1.1 Modelli AR per la simulazione del rumore

Generalmente, il rumore EEG di fondo viene simulato utilizzando dei modelli ARMA. Applicando in ingresso ad un modello con ordine e coefficienti adeguati, una sequenza di rumore bianco si ottiene in uscita un segnale EEG indistinguibile da quello originale [41].



Figura 5.1.1: Esempio di un modello ARMA in cui $u(n)$ è il rumore bianco d'ingresso che produce in output il segnale $y(n)$ [41]

Nel caso in esame si impiegano dei modelli autoregressivi (AR), particolari modelli AR-

MA, la cui equazione alle differenze è la seguente:

$$y_n = \sum_{k=1}^p a_k y_{n-k} + u_n \quad (5.1.1)$$

dove $y(n)$ è la serie sotto investigazione, a_k sono i coefficienti del modello AR, p è l'ordine del modello, u_n è il rumore bianco a media nulla e varianza δ_u^2 . Noto l'ordine p , identificare il modello AR significa calcolare i coefficienti a_k e la varianza δ_u^2 . La serie $y(n)$ deve godere di precise proprietà: deve essere un processo lineare, stazionario ed a media nulla. Per simulare EEG di fondo tramite modelli AR c'è bisogno di parametri adeguati. Per ottenere questi parametri si identificano i modelli AR su porzioni di EEG in cui non si è verificato nessun evento correlato allo stimolo. Inoltre, per soddisfare l'ipotesi di stazionarietà, vengono considerate piccole parti di EEG inferiori ai due secondi.

Nel caso specifico, la porzione lineare e stazionaria del segnale EEG usata è la parte di pre-stimolo, ovvero la parte della sweep fra -500 ms e 0 ms, alla quale viene sottratta la *baseline* (in modo che il segnale abbia media nulla). Chiaramente, è indifferente l'uso del pre-stimolo delle sweeps target o delle non target, proprio perché in entrambi i casi questa è la parte esente dal fenomeno fisiologico di occorrenza della P300.

Per ogni pre-stimolo si calcolano i modelli AR dall'ordine 2 al 14. Fra questi si sceglie quello con ordine ottimo secondo il criterio Akaike Information Criterion (AIC) [41]. Infine vengono considerati e salvati i soli modelli che superano il test di bianchezza (il rumore deve essere bianco) e che sono stabili, cioè con tutti i poli all'interno del cerchio di raggio unitario, concetto correlato alla stazionarietà del meccanismo di generazione dei dati. Per il test di bianchezza è stato utilizzato un test di Anderson con livello di significatività pari al 5%, mentre la correlazione è stata calcolata su 25 campioni.

5.1.1.2 Generazione delle sweeps non target

Le sweeps non target vengono generate come processi AR utilizzando i parametri precedentemente salvati. Il procedimento prevede di creare un vettore di N campioni provenienti da una distribuzione normale random.

Questi vengono moltiplicati per la standard deviation δ_u del rumore bianco stimata.

Facendo riferimento alla figura 5.1.1, il vettore così descritto, definisce il segnale d'ingresso $u(n)$. A questo vettore sarà applicato il filtro che ha per coefficienti gli a_k del modello AR selezionato. Per il filtraggio si applica la funzione Matlab *filter*. Il segnale di output $y(n)$ così ottenuto è il rumore EEG simulato, nonché la sweep non target.

Per garantire una buona variabilità, i parametri del modello AR nonché la varianza del rumore bianco d'ingresso vengono selezionati in modo casuale dall'insieme creato come indicato nel paragrafo 5.1.1.1.

Il valore di N , dimensione del vettore rappresentante il rumore bianco, sarà pari a 400 campioni. I primi 100 verranno scartati per limitare gli effetti del transitorio iniziale, gli altri 300 campioni costituiscono la sweep vera e propria.

5.1.2 Simulazione delle sweeps target

La seconda tipologia di sweeps da simulare è quella delle target. La traccia registrata può essere pensata come la somma di due componenti: da un lato la P300 (parte del segnale d'interesse); dall'altro la restante componente del segnale EEG che può essere interpretato come rumore. In altri termini, la sweep $y(t)$ può essere scritta come segue

$$y(t) = P300(t) + EEG(t) \quad (5.1.2)$$

Per simulare il segnale $y(t)$ sarà necessario simulare due componenti

1. $EEG(t)$: è la componente di rumore, per cui il processo di simulazione è identico a quello delle sweeps non target descritto nella sezione 5.1.1;
2. $P300(t)$: è la componente relativa unicamente all'onda P300, la cui costruzione prevede di crearne una template (sezione 5.1.2.1) e simulare una considerevole variabilità (sezione 5.1.2.2).

5.1.2.1 Costruzione della template

La template viene ottenuta sfruttando le sweeps a disposizione. In particolare, fra tutti i dati dei soggetti disponibili, si è scelto il soggetto M5 poiché le sweeps presentano la componente P300 più rappresentativa. Anche il pre-stimolo per la simulazione del rumore è stato prelevato dal soggetto M5. Le sweeps utilizzate sono le *fssa2*, quindi solo le sweeps target filtrate a 15 Hz e con rimozione dell'artefatto. Inoltre, viene implementato il filtro passa alto con frequenza di taglio ad 1 Hz. Esso ha lo scopo di eliminare trend lineari che si possono trovare in numerose tracce. E' stato implementato un filtro butterworth di ordine quattro ed è stata utilizzata la funzione Matlab *filtfilt*. Il modulo della risposta in frequenza e della fase sono rappresentati in figura 5.1.2.

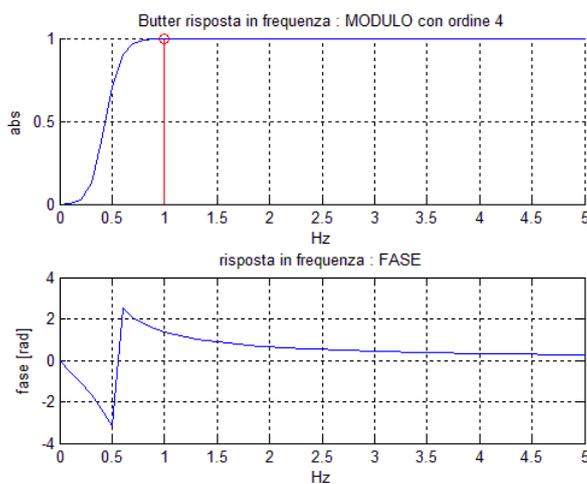


Figura 5.1.2: *Modulo e fase della risposta in frequenza del filtro passa alto a 1 Hz.*

Il numero di sweeps target a disposizione è pari a 175. Ogni sweep è stata così filtrata ed incolonnata in una matrice, la quale alla fine avrà dimensione 300x175.

Infine, è stata fatta la media rispetto al tempo ottenendo un vettore di 300 campioni rappresentato in figura 5.1.3.

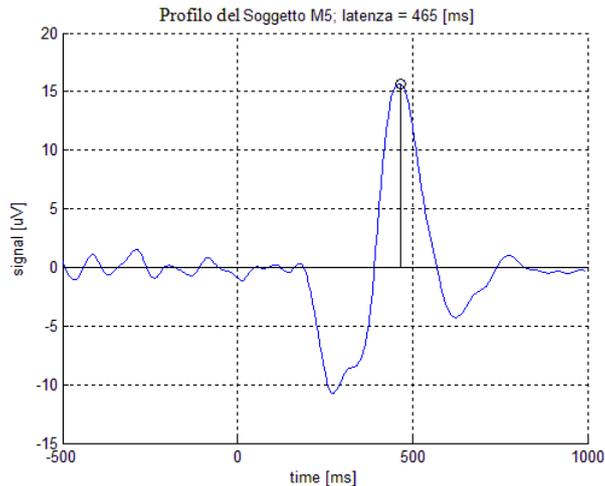


Figura 5.1.3: *Profilo ottenuto mediando le sweeps target pre-filtrate del soggetto M5.*

Al profilo così ottenuto è stata tolta la *baseline*, ovvero la media del segnale calcolata nella parte di pre-stimolo. Si nota come effettivamente la P300 sia completamente distinta dal resto del segnale. Essa ha un'ampiezza molto pronunciata pari a circa 15 μV ed una latenza pari a 465 ms. E' stato considerato il segnale nell'intervallo temporale fra 90 ms a 740 ms ed esso è stato traslato in modo che la latenza della template sia a 300 ms. La template ottenuta è rappresentata in figura 5.1.4.

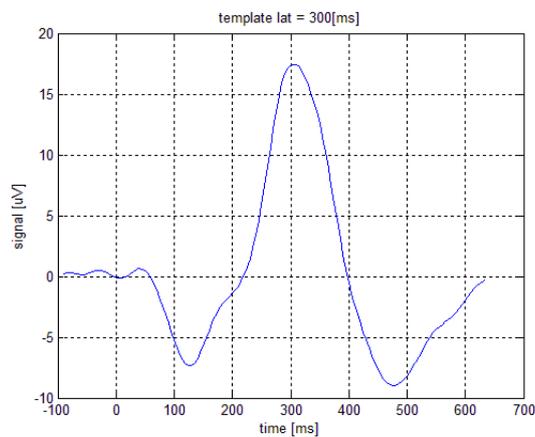


Figura 5.1.4: *Template del soggetto M5.*

5.1.2.2 Generazione delle sweeps target

Per generare le sweeps target è necessario, come già detto, simulare una variabilità della componente P300. Il procedimento prevede di estrarre in modo random un valore di latenza ed un valore di ampiezza e, secondo tali valori, traslare ed amplificare la template. In particolare, l'ampiezza è stata fatta variare nel range $[10,20] \mu V$, mentre la latenza è stata fatta variare nel range $[-100, 100]$ ms]. In entrambi i casi i valori sono uniformemente distribuiti.

In figura 5.1.5 si mostra un esempio della variabilità delle componenti P300 simulate, rappresentate in blu, rispetto alla template originale, di colore rosso.

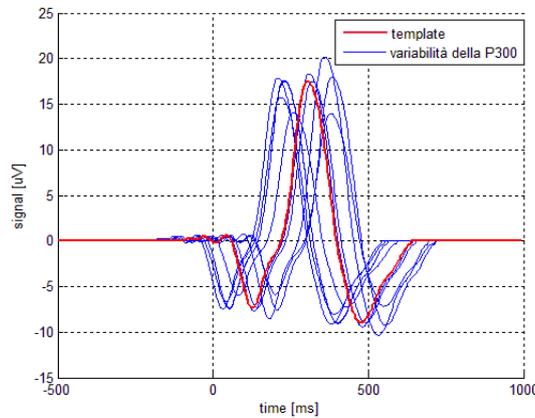


Figura 5.1.5: Variabilità della sola componente P300 generata. In rosso è la template originale, mentre in blu si rappresentano degli esempi di P300 ottenute dalla template in seguito a variazioni della latenza e dell'ampiezza.

I valori dell'ampiezza e della latenza della $P300(t)$ generata sono noti e memorizzati durante il processo di simulazione. In questo modo, è possibile calcolare il rapporto segnale-disturbo (SNR).

Le definizioni di SNR in letteratura sono molteplici. In questo lavoro si definisce SNR il rapporto fra la potenza del segnale $P300(t)$ e la potenza del segnale $EEG(t)$.

In formule

$$SNR = \frac{\sum_t (|P300(t)|^2)}{\sum_t (|EEG(t)|^2)} \quad (5.1.3)$$

Mentre i valori di SNR sono utili per comprendere la rumorosità delle sweeps simulate, i valori veri di latenza ed ampiezza verranno usati in seguito per il confronto diretto con quanto stimato dall'algoritmo Bayesian Two Step e dal Multi Task Learning.

5.1.3 Scelta del pre-stimolo per la simulazione dei dati

Nei paragrafi precedenti si è descritto il processo di simulazione delle sweeps, il tutto a partire dall'identificazione del modello AR dal prestimolo selezionato. In questo senso, l'unica precisazione fatta in precedenza riguardava il fatto che è indifferente l'uso del pre-stimolo delle tracce target o non target, proprio perché in entrambi i casi tale parte non è coinvolta nella generazione della componente P300.

L'aspetto che adesso viene introdotto riguarda invece l'elaborazione delle sweeps. Come descritto nella sezione 2.2.2, le tracce possono essere *grezze* (*ss2*), oppure elaborate con un filtro a 15 Hz (*fssa2*). E' possibile scegliere di simulare i dati utilizzando i modelli AR identificati dal pre-stimolo dei dati grezzi (caso I) o dal pre-stimolo dei dati filtrati a 15 Hz (caso II).

5.1.3.1 Caso I: prestimolo estratto da dati grezzi

Le sweeps target e non target vengono simulate sulla base del pre-stimolo delle sweeps target grezze, ovvero le *ss2*. Il numero di tali sweeps è pari a 254. Per ogni pre-stimolo si identificano i modelli AR e si scelgono fra questi quelli bianchi e stabili che sono in numero pari a 170 secondo la procedura descritta.

In figura 5.1.6 è riportato l'istogramma dell'SNR delle sweeps simulate target.

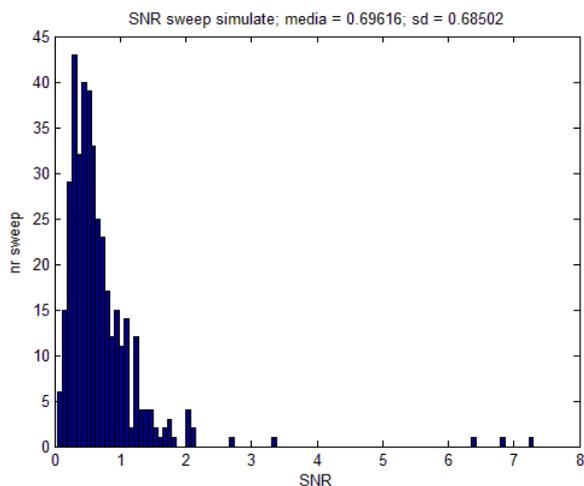


Figura 5.1.6: Iistogramma del SNR delle sweeps simulate con pre-stimolo grezzo.

Si evidenzia subito come il valore medio di SNR sia pari a 0.7. Questo valore può essere normalmente ritrovato su tracciati EEG reali.

In figura 5.1.7 sono riportati degli esempi di sweeps target e non target simulate. La curva in blu fa riferimento alle sweeps target, quella in rosso fa riferimento alle sweeps non target. Si nota subito come queste sweeps siano nettamente rumorose, al punto tale che la componente P300 è praticamente non distinguibile dal rumore di fondo nella

maggior parte delle sweeps, come ad esempio nella prima sweep in alto a sinistra.

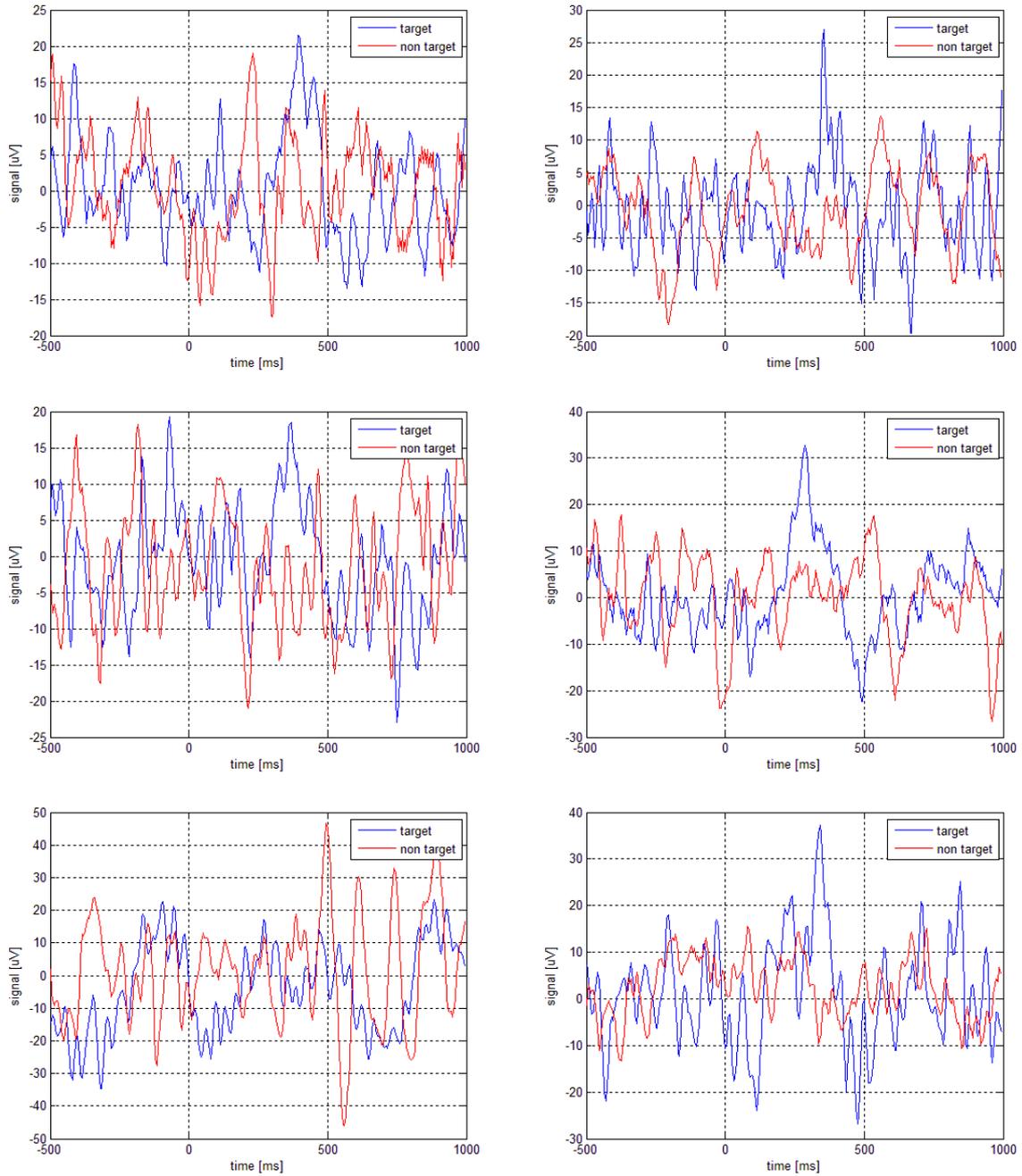


Figura 5.1.7: *Sweeps target (in blu) e non target (in rosso) simulate con pre-stimolo grezzo.*

5.1.3.2 Caso II: prestimolo da sweeps filtrate a 15 Hz

Un'altra possibilità consiste nell'identificare i modelli AR utilizzati per la simulazione del rumore da tracce target filtrate a 15 Hz e con rimozione dell'artefatto (*fssa2*). Il numero di sweeps target è pari a 175. Per ogni pre-stimolo si identificano i modelli AR e, fra questi, quelli stabili e bianchi sono risultati in numero pari a 104.

Le tracce sono state simulate secondo quanto descritto nei paragrafi precedenti. In figura 5.1.8 si riporta l'istogramma del SNR.

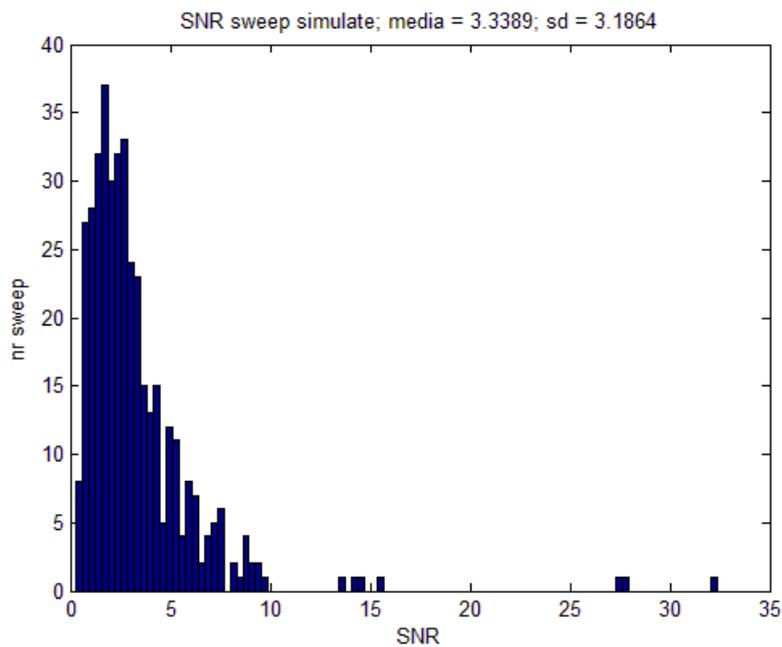


Figura 5.1.8: *Istogramma delle sweeps target simulate mediante pre-stimolo filtrato passabasso a 15 Hz.*

Come si può vedere, la media del SNR è pari a 3.34, un SNR molto favorevole e anche non consueto proprio perché in generale le tracce sono ben più rumorose.

In figura 5.1.9 sono riportati esempi di sweeps simulate. In blu sono raffigurate le sweeps target, in rosso quelle non target. E' evidente l'effetto di un SNR così favorevole. La componente P300 si riesce a distinguere chiaramente dal resto del segnale, come ad esempio per la sweep del pannello in alto a sinistra.

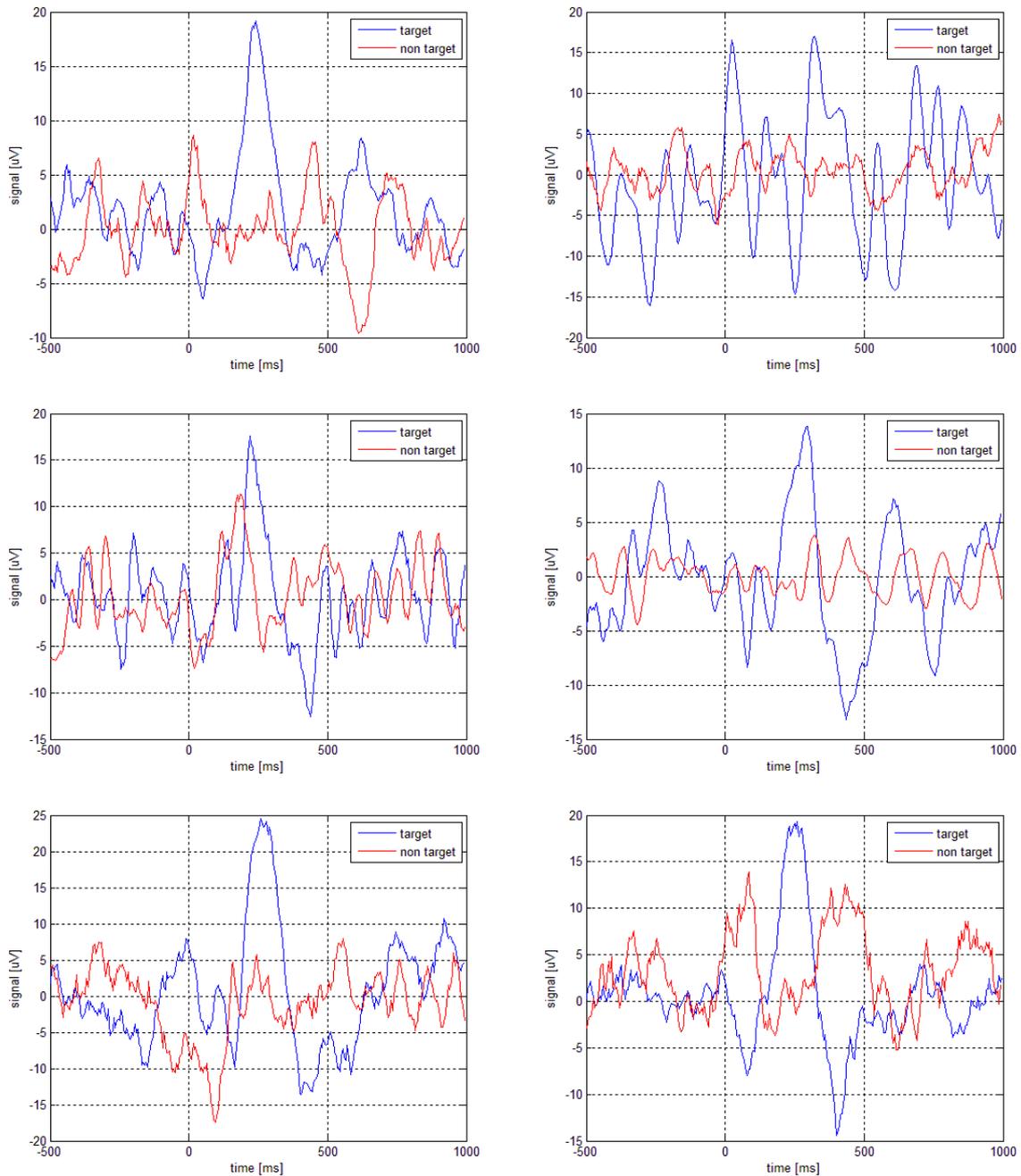


Figura 5.1.9: *Sweeps target (in blu) e non target (in rosso) simulate con pre-stimolo filtrato a 15 Hz.*

5.1.3.3 Considerazioni finali

Come si nota nelle figure 5.1.7 e 5.1.9, la principale differenza nei due casi analizzati sta proprio nella natura del rumore. Si nota come nelle sweeps simulate nel caso II, la componente P300 sia ben distinguibile dal resto del segnale, mentre nel caso I il rumore copre

la componente nella maggior parte dei casi. Poiché il fine dei dati simulati è appunto simulare il segnale vero, si comprende come basare l'analisi sulle tracce del caso II potrebbe falsificare i risultati: le eventuali buone performance potrebbero dipendere pesantemente dall'elevato SNR. Il caso I invece propone delle sweeps più vicine alla realtà. In questo modo si può realmente testare la capacità dell'algorithmo di estrarre la componente P300 qualora essa sia non riconoscibile a causa del basso SNR. Per questi motivi, sono stati scelti i pre-stimoli delle sweeps grezze per estrarre i parametri dai modelli AR usati per simulare il rumore EEG.

5.2 Metodo B2S: risultati

In questa sezione, verranno mostrati i risultati della stima ERP ottenute applicando il metodo bayesiano a due step alle sweeps target simulate, in numero pari a 80. Poiché il segnale vero $P300(t)$, nonché la sua latenza ed ampiezza, sono note, è possibile confrontare direttamente quanto la stima si avvicina al caso vero.

In figura 5.2.1 è mostrato un esempio di stima dell'ERP medio. Si precisa che la P300 è stata identificata come il massimo del profilo stimato. La media delle P300 vere è rappresentata in linea verde, la media ottenuta dall'averaging convenzionale in linea blu e la media stimata in linea rossa.

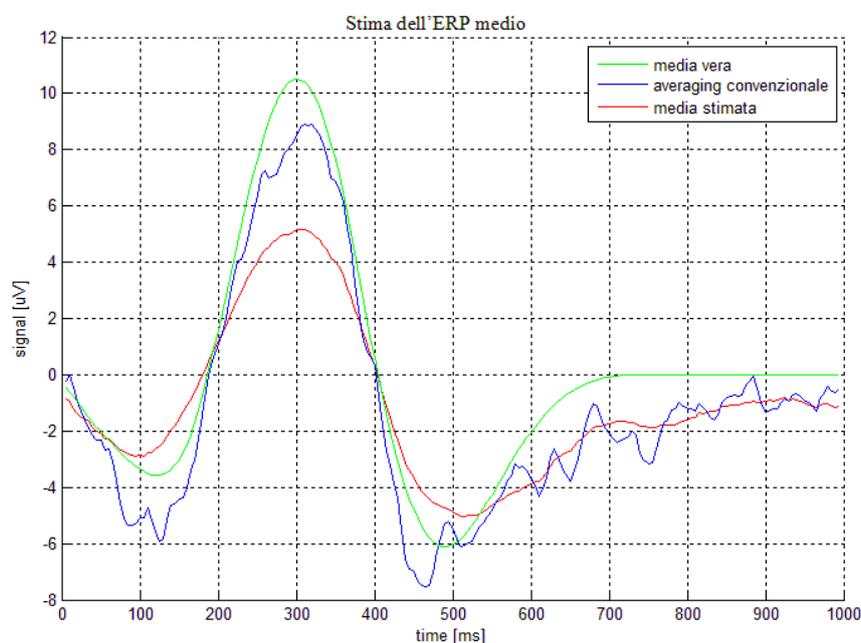


Figura 5.2.1: *Stima dell'ERP medio dai dati simulati con il metodo B2S. La media vera in linea verde, la media convenzionale in blu e la media stimata in rosso.*

Dalla figura 5.2.1 è evidente come la media ottenuta mediante averaging convenzionale sottostimi la media vera. Inoltre, la stessa media stimata con B2S risulta essere ancora più bassa in ampiezza. Ciò fa supporre che le stime soffrano nell'identificazione della corretta ampiezza della componente ERP.

In figura 5.2.2 sono riportate alcuni esempi di stime single trial.

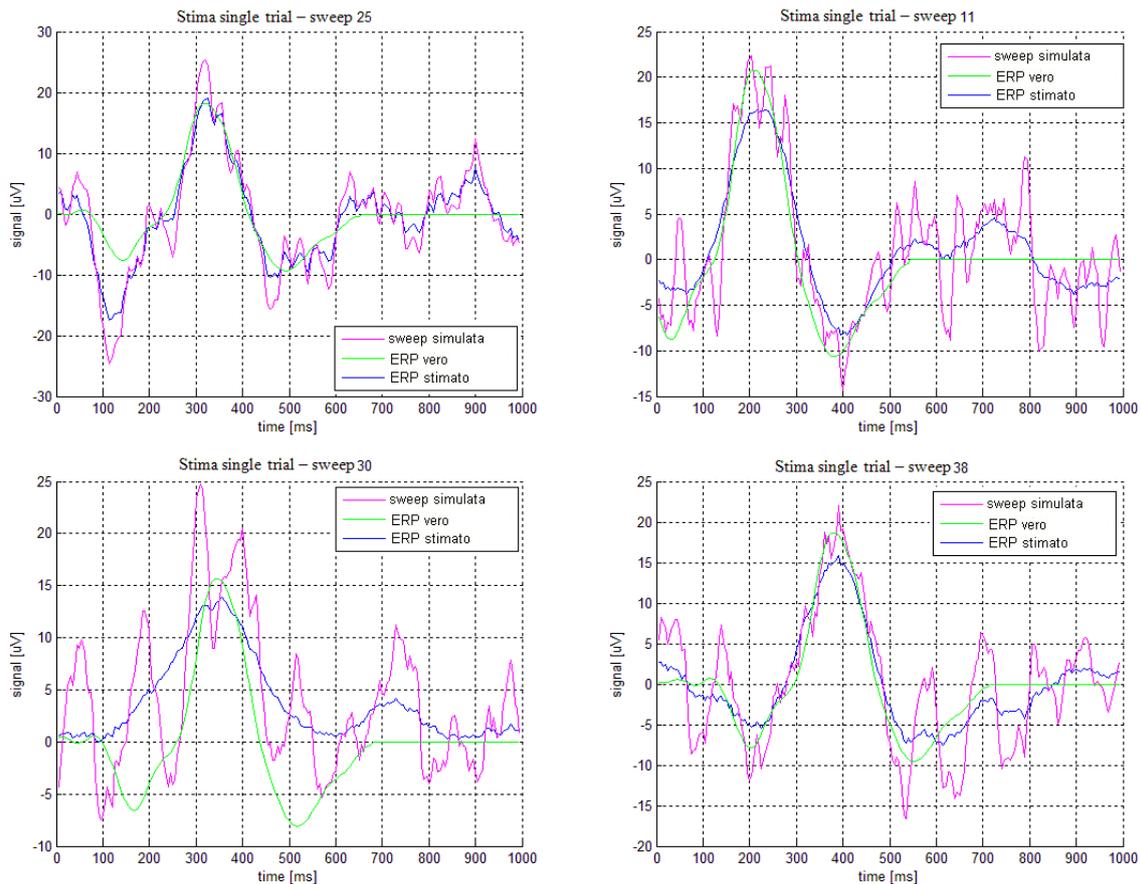


Figura 5.2.2: Esempi di stime single trial ottenute con il metodo B2S. In magenta la sweep simulata, in verde la P300 vera ed in blu la stima B2S.

In magenta è riportata la sweep simulata (grezza), in verde la P300 vera che l'ha generata ed infine in blu la stima B2S ottenuta.

E' evidente che i risultati sono molto buoni: i profili ottenuti con il metodo B2S sono abbastanza simili alla componente vera, fornendo delle stime molto precise in termini di latenza. Nei grafici di figura 5.2.2, anche l'ampiezza sembra esser stimata bene. In realtà, questo non sempre accade ed in alcuni casi l'ampiezza stimata è molto lontana dal valore vero. Si descriveranno nel dettaglio questi casi nella sezione successiva.

In figura 5.2.3 sono riportati dei grafici di correlazione tra latenze ed ampiezze vere ed i corrispondenti valori stimati. Sull'asse delle ascisse sono riportati i valori veri, su quella delle ordinate i valori stimati. L'obiettivo è valutare come questi campioni si distribuis-

cono sulla retta a 45°, indicata in rosso. Tanto più questi saranno vicini a tale retta, tanto più le stime calcolate saranno corrette.

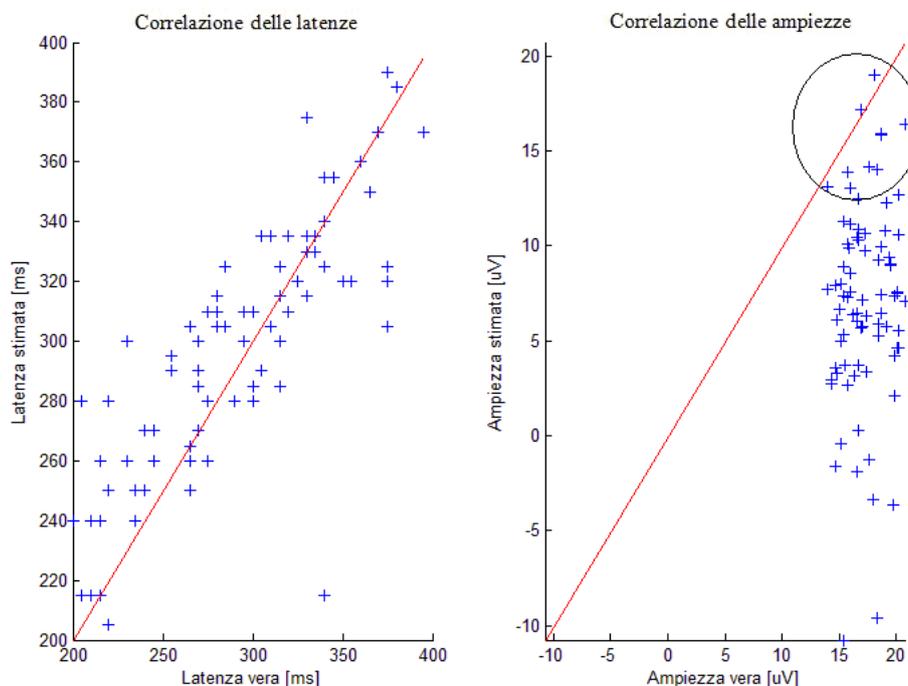


Figura 5.2.3: Correlazione delle latenze (sx) e delle ampiezze (dx) stimate con il metodo B2S rispetto quelle vere.

Come si nota dal grafico 5.2.3, la correlazione delle latenze è abbastanza buona. Anche se si nota che in alcuni casi l'errore di stima è elevato, in generale i valori sono abbastanza allineati alla retta rossa.

Per l'ampiezza invece si conferma quanto detto in precedenza. Solo la parte compresa nel cerchio nero si avvicina alla retta a 45°, mentre i restanti campioni sono decisamente distanti dalla stima vera. Inoltre tutte le stime errate dell'ampiezza forniscono un valore inferiore rispetto a quello vero.

I valori di correlazione sono stati calcolati con la function Matlab *corrcoef*. La formula a cui si fa riferimento è la seguente

$$R(i, j) = \frac{c(i, j)}{\sqrt{c(i, i) * c(j, j)}} \quad (5.2.1)$$

in cui c è la covarianza di due generici vettori X ed Y. Nel caso in esame, si fa riferimento al vettore dei valori stimati e a quello dei valori veri.

Accanto a questo valore, viene fornito il p -value, che indica quanto significativo è il valore di correlazione trovato: più basso è, per esempio inferiore allo 0.05, più importante è la correlazione.

	correlazione	p-value
latenza	0,8125	$6,06e^{-20}$
ampiezza	0,1425	0,21

Tabella 5.1: *Valore di correlazione e relativo p-value per la latenza e per l'ampiezza stimata dal metodo B2S.*

Quanto detto viene confermato anche dai valori di correlazione ottenuti. La latenza ha un ottimo valore di correlazione con un corrispondente p-value molto basso a conferma della sua significatività. Mentre l'ampiezza ha un valore di correlazione molto basso ed il p-value elevato. Ciò conferma quanto detto in precedenza.

In figura 5.2.4 sono infine riportati gli istogrammi delle latenze vere e stimate, per confrontare eventuali differenze nelle distribuzioni dei valori.

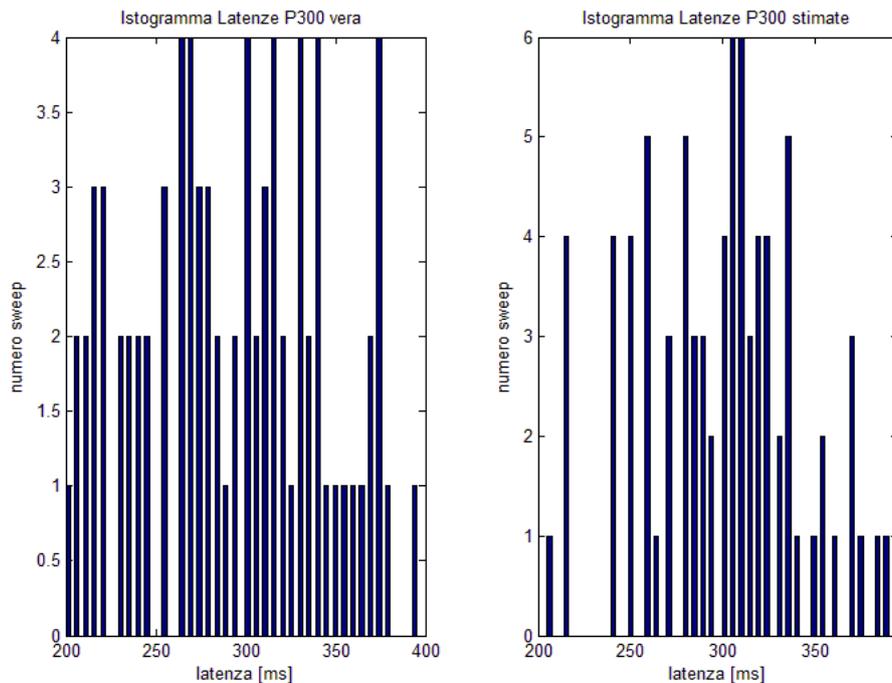


Figura 5.2.4: *Istogramma delle latenze vere (sx) e delle latenze stimate mediante il metodo B2S (dx).*

Anche con questo grafico si possono apprezzare le buone stime della latenza. Il range di variazione dei valori stimati, nonché la distribuzione dei valori all'interno dell'intervallo, riproducono il comportamento delle latenze vere. La situazione per le ampiezze è sicuramente diversa, si confrontino a tal proposito gli istogrammi mostrati in figura 5.2.5.

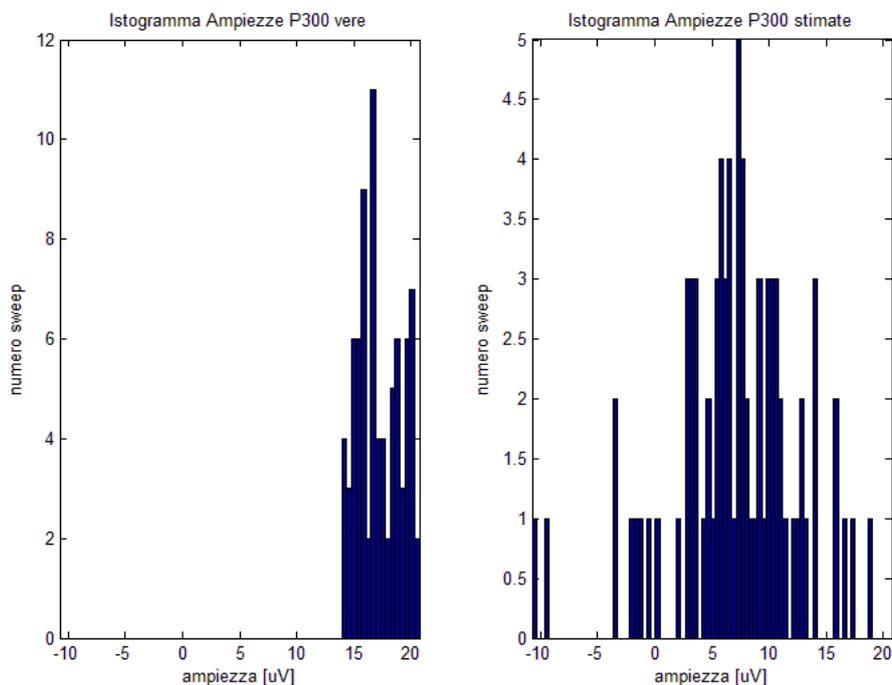


Figura 5.2.5: *Istogramma delle ampiezze vere (sx) e delle ampiezze stimate mediante il metodo B2S (dx).*

Si nota infatti come il range dell'istogramma delle ampiezze stimate contiene quello delle ampiezze vere. In particolare, i valori stimati sono soprattutto distribuiti verso i valori più bassi di ampiezze, a conferma dell'evidente sottostima.

5.2.1 Casi particolari

Nel precedente paragrafo, sono state messe in luce problematiche della tecnica B2S relativamente alla stima dell'ampiezza della componente P300.

Si considerano a tal proposito alcuni esempi riportati in figura 5.2.6.

La legenda è la medesima: in magenta la sweep simulata, in verde la P300 vera, in blu la stima B2S. E' evidente come l'ampiezza della componente P300 venga sottostimata. In particolare, si sono voluti riportare due casi in cui il segnale grezzo già interpreta di suo la componente ERP, ma l'algoritmo sembra proprio non seguire questa informazione. L'aspetto a favore di queste stime è che la latenza della componente ERP viene ancora correttamente stimata.

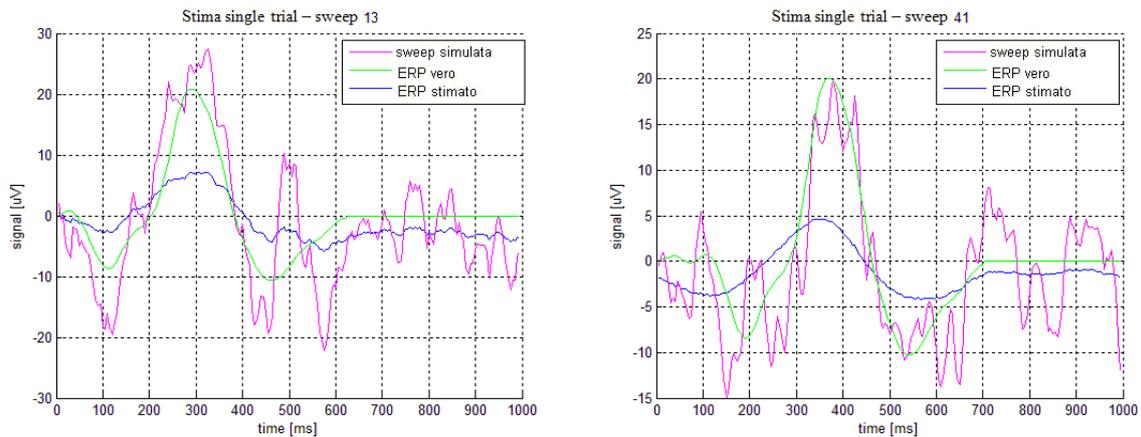


Figura 5.2.6: Esempi di stime single trial con evidente sottostima dell'ampiezza.

Nell'analisi delle tracce si sono individuati casi in cui la componente P300 non viene stimata bene, ma parlare, di sottostima dell'ampiezza può essere fortemente limitativo. Si considerino le tracce mostrate in figura 5.2.7.

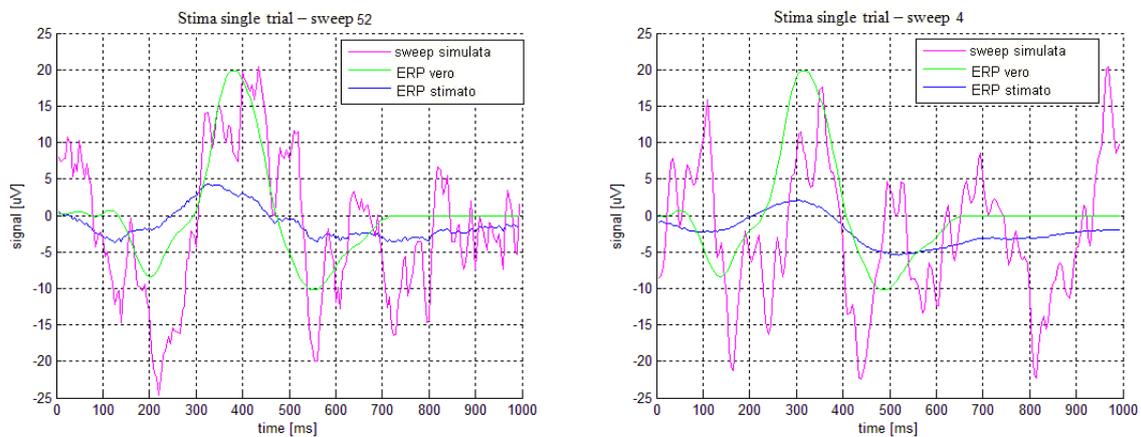


Figura 5.2.7: Esempi di stime single trial con il metodo B2S affette da oversmoothing.

Le stime ottenute sono molto regolari al punto che sembra molto limitativo far rientrare questi casi in quelli di sottostima dell'ampiezza. Questi fenomeni sembrano più legati ad un aspetto introdotto nel capitolo 3: l'*oversmoothing* delle sweeps. Il vettore di riferimento μ è calcolato sulla base del prior, informazione a priori, fornito all'algoritmo. Nel caso specifico, esso è stato calcolato dalla media pesata della stima delle sweeps con vettore di riferimento nullo. Quando tale informazione non è sufficiente, la stima B2S risulta smooth.

5.3 Metodo MTL: risultati

In questa sezione si affronta l'analisi delle stime MTL dalle tracce target simulate, in numero pari a 80. In figura 5.3.1 è mostrato il risultato per quanto concerne la stima dell'ERP medio. La curva in blu è la stima fornita dalla tecnica di averaging convenzionale, quella verde fa riferimento alla media vera delle sweeps $P300(t)$ ed infine la rossa è la media stimata, ovvero la curva di popolazione.

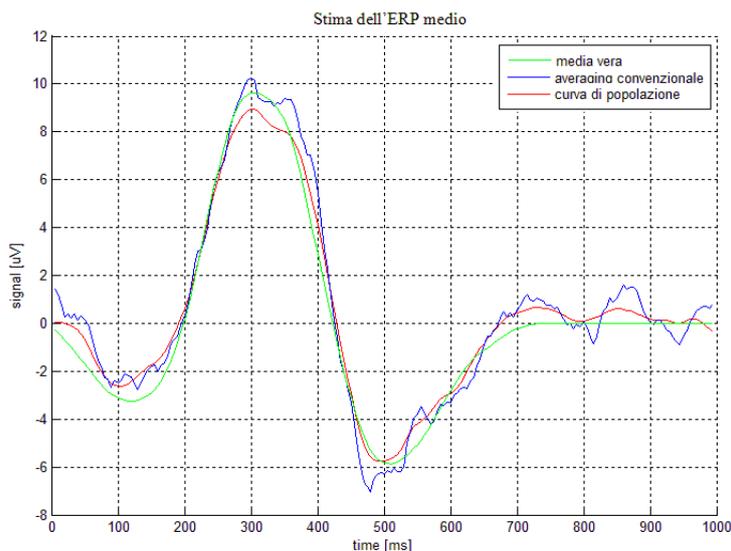


Figura 5.3.1: *Stima dell'ERP medio dai dati simulati con il metodo MTL. La media vera è in linea verde, la media con averaging convenzionale in linea blu e la curva di popolazione in linea rossa.*

Dalla figura è evidente come la stima fornita dal metodo MTL sia molto simile alla curva vera. Si nota inoltre il miglioramento rispetto alla stima fornita dalla tecnica di media convenzionale che è caratterizzata da picchi spuri non presenti nel segnale vero. Anche le stime single-trial sono soddisfacenti.

La figura 5.3.2 mostra degli esempi di stime ottenute per le singole sweeps. La sweeps simulata è rappresentata da una linea magenta, l'ERP vero da una linea verde, l'ERP stimata da una linea blu. Infine, in rosso è mostrata la curva di popolazione.

Si noti come la stima individuale sia in grado di distinguere la componente P300 dal resto del segnale, anche quando quest'ultima non è d'immediata individuazione (come nel caso della sweeps 10, pannello in alto a destra).

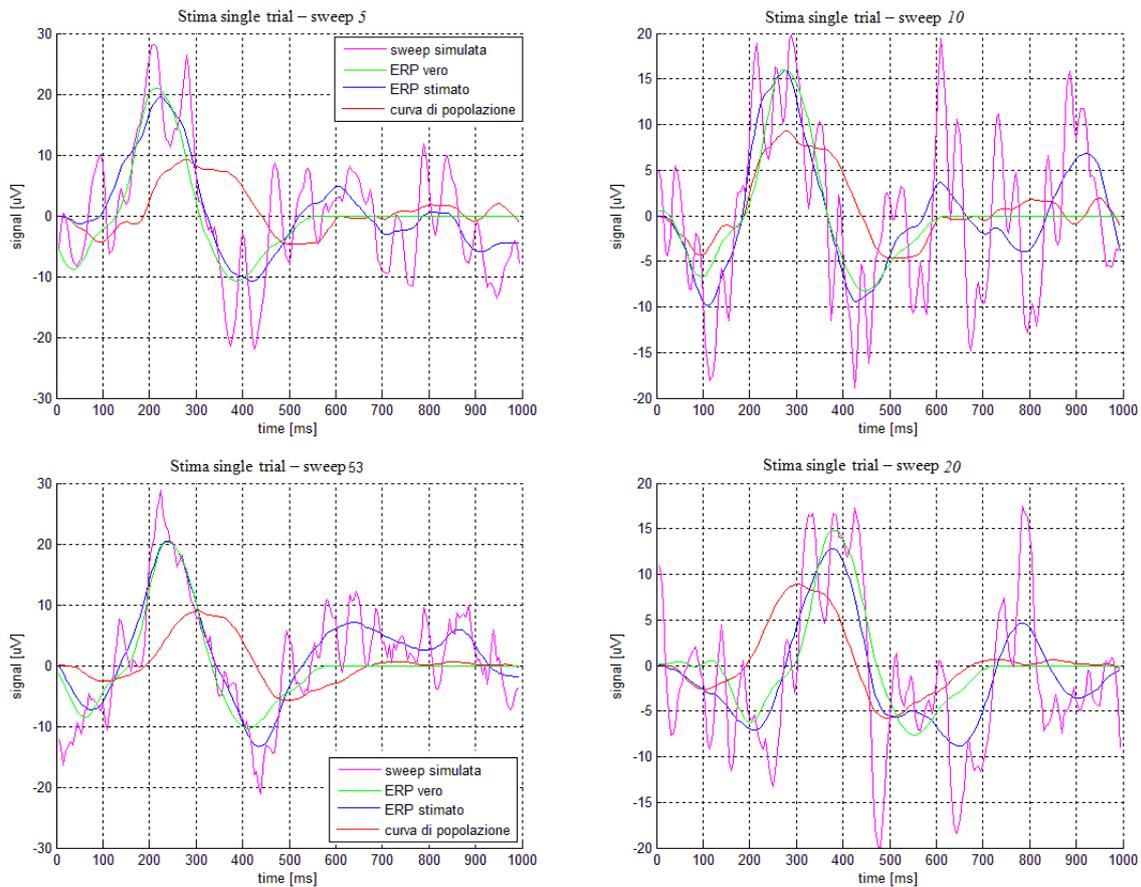


Figura 5.3.2: Esempi di stima single-trial fornita dal metodo MTL. In magenta la sweep simulata, in verde la P300 vera, in blu la stima MTL ed in rosso la curva di popolazione.

Come per il metodo B2S, anche per il metodo MTL la latenza viene sicuramente meglio stimata rispetto all'ampiezza. In figura 5.3.3 sono riportati i grafici di correlazione tra valori veri e stimati per le latenze (pannello a sinistra) e le ampiezze (pannello a destra).

Come è evidente dalla figura, per la latenza si ottengono degli ottimi risultati, con valori distribuiti lungo la retta a 45° .

Per quanto concerne l'ampiezza, la situazione sembra non proprio ottimale. Sembra esserci una buona correlazione nella parte centrale, ma il numero di campioni al di sotto e al di sopra della retta a 45° è elevato. In particolare sembrano presenti in numero maggiore le ampiezze sottostimate. I valori di correlazione ottenuti sono indicati in tabella 5.2.

Si nota che il valore di correlazione della latenza è molto elevato con un relativo valore del p-value molto basso a conferma della significatività del risultato. Il valore di correlazione dell'ampiezza è basso, con il relativo p-value elevato. Questo risultato conferma quanto detto in precedenza.

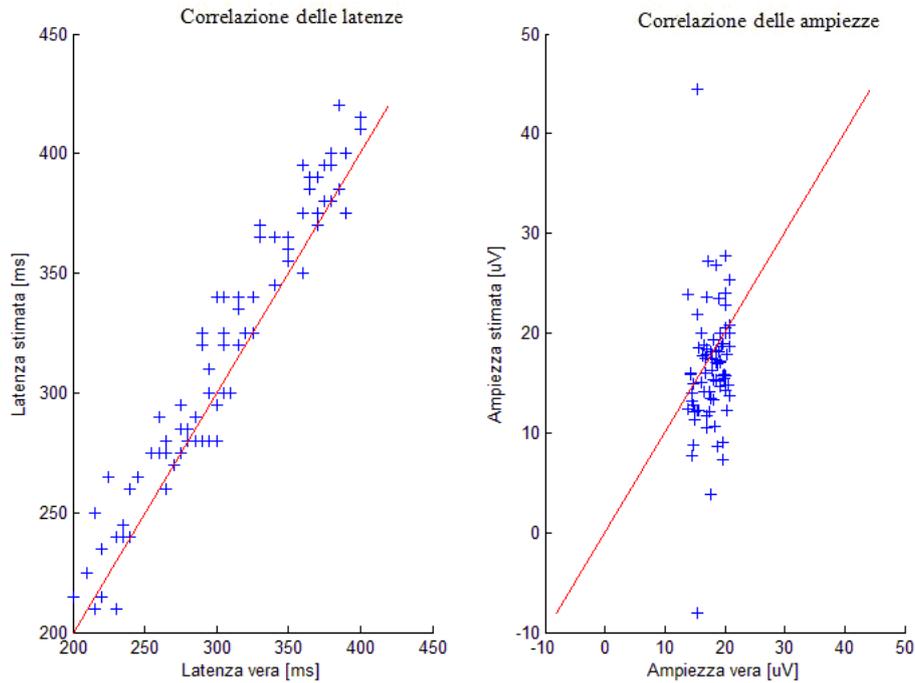


Figura 5.3.3: Grafico di correlazione della latenza (sx) e dell'ampiezza (dx).

	correlazione	p-value
latenza	0,9690	$3,95e^{-49}$
ampiezza	0,1792	0,11

Tabella 5.2: Valore di correlazione e relativo p-value per la latenza e per l'ampiezza.

Infine, sono riportati i grafici degli istogrammi delle latenze e delle ampiezze in figura 5.3.4 e 5.3.5. Per quanto concerne l'istogramma delle latenze, la distribuzione delle latenze vere è fra i 200 ms ed i 400 ms e quindi si può affermare che il range di variabilità delle latenze della P300 è stato stimato correttamente. Anche la distribuzione dei valori all'interno dell'intervallo è confrontabile.

Per quanto riguarda l'istogramma delle ampiezze, si conferma il fatto che l'algoritmo "soffre" nella stima delle ampiezze. Il range è molto più ampio nel caso delle ampiezze stimate rispetto a quelle vere a conferma della sovrastima e sottostima dei valori.

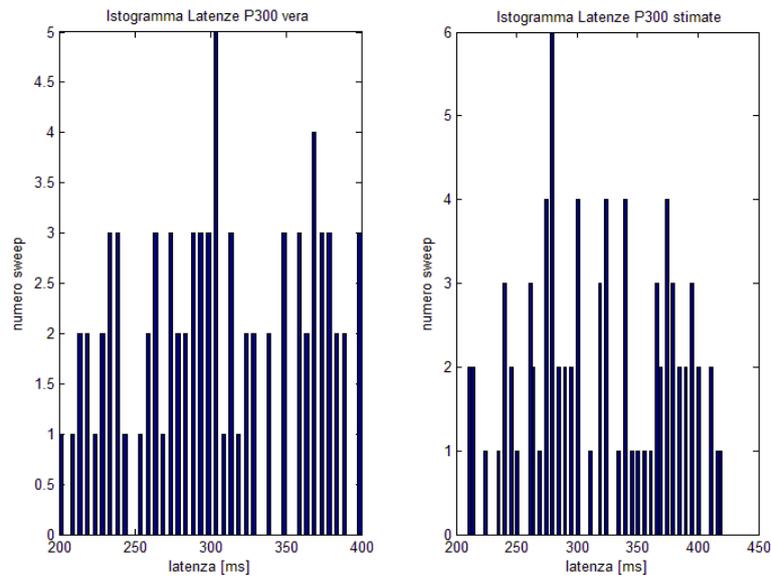


Figura 5.3.4: *Istogramma delle latenze vere (sx) e delle latenze stimate (dx).*

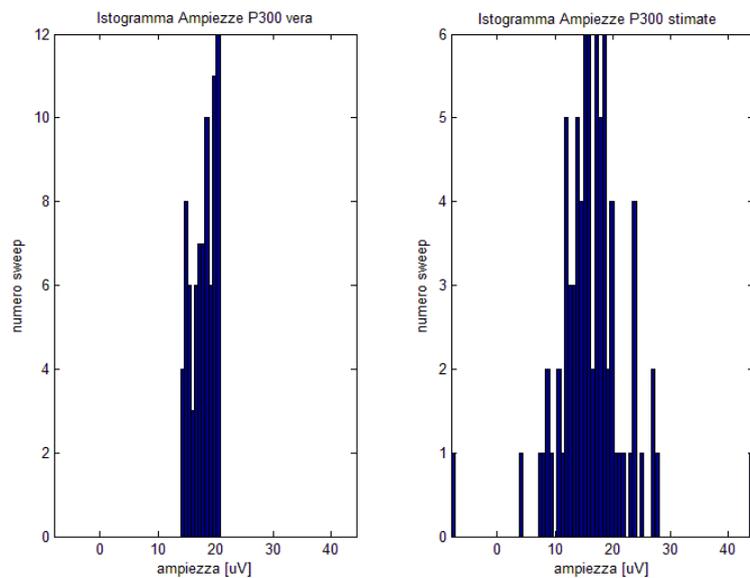


Figura 5.3.5: *Istogramma delle ampiezze vere (sx) e delle ampiezze stimate (dx).*

5.3.1 Casi particolari

Nella visualizzazione delle stime MTL dai dati simulati non si sono riscontrati evidenti casi particolari. E' ancora presente la problematica relativa all'ampiezza, ma non in modo evidente come nel caso del metodo B2S. In particolare, il metodo MTL non sembra polarizzato nella stima dell'ampiezza della P300. In figura 5.3.6 vengono riportati comunque

dei casi in cui la stima dell'ampiezza della P300 non è ottimale.

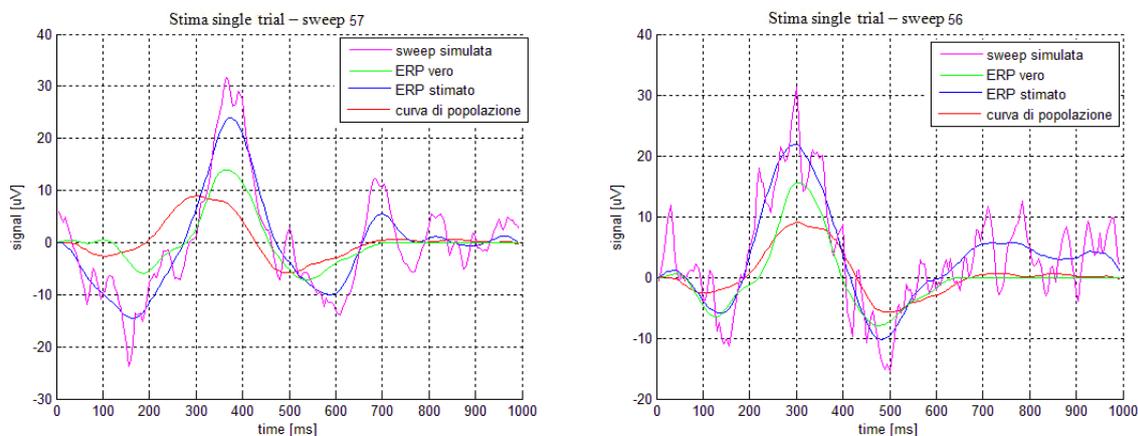


Figura 5.3.6: Esempi di stime single trial con sovrastima dell'ampiezza della P300.

La sovrastima del segnale è dovuta al fatto che la stima MTL segue il segnale originale, facendone un filtraggio locale, senza però distaccarsi da questo in modo da poter stimare la vera componente P300. Chiaramente, non si può non notare che la latenza della P300 è identificata correttamente. Nel caso invece delle sottostima, si propongono i seguenti esempi in figura 5.3.7. In questo caso invece, la stima non è in grado di raggiungere il

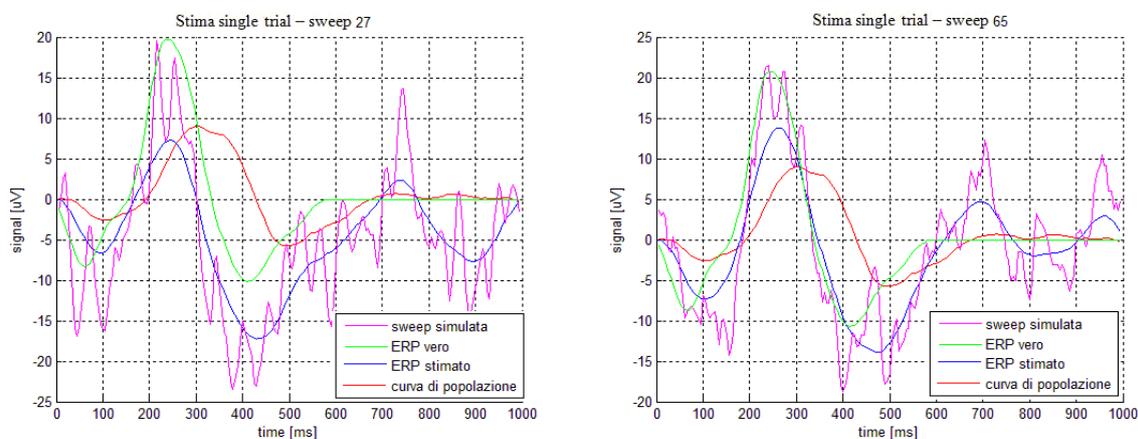


Figura 5.3.7: Esempi di stime single trial con sottostima dell'ampiezza della P300.

picco della P300 vera, nonostante esso fosse comunque riconoscibile dalla curva grezza. Si nota, invece, ancora una volta la buona stima della latenza.

5.4 Conclusioni

Nella prima parte del capitolo è stato descritto il processo di generazione dei dati simulati. Le sweeps target sono state ottenute sommando ad una componente di rumore EEG

una componente P300, mentre le sweeps non target sono state ottenute considerando la sola componente di rumore. Il rumore EEG è stato simulato come un processo AR i cui parametri sono stati ricavati da pre-stimoli estratti da dati reali. Le componenti ERP sono state ottenute, invece, tramite traslazione e amplificazione di una template ottenuta dalla media convenzionale stimata dalle sweeps registrate su un soggetto reale. Le sweeps non target hanno così la sola componente EEG di rumore.

Nella seconda parte invece sono stati presentati i risultati ottenuti con il metodo B2S e con il metodo MTL. Il grande vantaggio di questo studio di simulazione sta nella possibilità di confrontare il profilo stimato con il profilo vero, nonché le latenze e le ampiezze stimate per la componente P300 con quelle reali.

In generale, sembra che le stime fornite dal metodo MTL sono migliori, proprio perché non c'è l'effetto di oversmoothing di cui sono affette le stime B2S. Entrambi i metodi stimano comunque bene la latenza della P300, parametro clinico di grande interesse, mentre soffrono nella stima dell'ampiezza. Il metodo B2S, in particolare, sembra essere negativamente polarizzato nello stimare l'ampiezza della componente P300.

Nel prossimo capitolo verranno mostrati i risultati relativi all'applicazioni di questi metodi a dati reali registrati durante sessioni BCI. Nel capitolo 7, invece, verrà valutata la performance del classificatore nel caso in cui siano impiegate queste stime come segnali da cui estrarre le features.

Capitolo 6

Applicazione delle tecniche B2S e MTL per la stima della P300 da dati reali

6.1 Costruzione del dataset

I dati utilizzati sono stati forniti dall'istituto IRCCS San Camillo. Il dataset è costituito dalle tracce acquisite su cinque soggetti affetti da ALS (indicati con la lettera M), e tre soggetti di controllo (indicati con la lettera S). Ogni soggetto ha una propria cartella in cui sono memorizzati i dati relativi alle diverse sessioni eseguite secondo il protocollo sperimentale 2.3.1. I dati utilizzati sono quelli acquisiti durante le sessioni di testing perché quelli più performanti.

Come detto nel capitolo 2, le sweeps sono già scisse in target, con l'acronimo $ss2$, e non target, con l'acronimo $ss1$. Il segnale così memorizzato è quello "grezzo" e può essere sottoposto a due diversi tipi di elaborazione

1. Filtraggio a 15 Hz e rimozione dell'artefatto sia per le sweeps target che non target, ottenendo così le tracce indicate rispettivamente con l'acronimo $fssa1$ e $fssa2$;
2. Rimozione dell'artefatto dalle sweeps grezze sia target che non target, ottenendo così le tracce indicate rispettivamente con l'acronimo $ssa1$ e $ssa2$.

Ad ogni soggetto verrà associata così una *struct* Matlab in cui sono memorizzate le quattro tipologie di sweeps ($ss1$, $ss2$, $fssa1$, $fssa2$, $ssa1$, $ssa2$).

Sia il metodo B2S che il MTL sono in grado di elaborare delle sweeps riferite ad un unico canale, è quindi necessario scegliere quale elettrodo si considera più rappresentativo fra quelli presenti nel protocollo di acquisizione (Fz, Cz, Pz, Oz). Visto quanto descritto nel capitolo 1, paragrafo 1.5.1, la massima ampiezza della P300 si manifesta in zona parietale, per cui sembra logico considerare le tracce registrate del canale Pz. Da qui in poi quindi, parlando di sweeps, si farà riferimento a quelle relative al canale Pz.

In tabella 6.1 è riportata la numerosità delle sweeps per ogni tipologia e per ogni soggetto. Si noti che, per la maggior parte dei soggetti, l'applicazione di un filtro passa-basso a 15 Hz e la successiva rimozione dell'artefatto è in grado di conservare un numero maggiore di sweeps rispetto alla procedura diretta di rimozione dell'artefatto (ad esempio si consideri il

soggetto M5). In alcuni casi, come per il soggetto S1, entrambe le elaborazioni ottengono lo stesso numero di sweeps.

sbj	n° ssl	n° ss2	n° fssal	n° fssa2	n°ssal	n° ssa2
M1	633	216	615	205	597	200
M2	432	151	387	137	377	135
M3	399	136	283	118	264	116
M4	359	119	216	111	209	110
M5	754	254	607	175	591	169
S1	335	118	327	118	327	118
S2	674	229	543	196	527	194
S3	702	242	553	202	546	200

Tabella 6.1: Numerosità delle sweeps per soggetto

6.2 Metodo B2S: risultati

In questa sezione verranno discussi i risultati ottenuti applicando il metodo B2S alle sweeps target *ssa2*. Chiaramente per questioni di spazio non si riportano i risultati di tutti i soggetti, ma solamente alcuni casi esemplificativi relativi a due soggetti di controllo, il soggetto S1 ed il soggetto S2, e due pazienti ALS, il soggetto M3 ed il soggetto M5. Si inizia l'analisi delle stime per i soggetti sani. In figura 6.2.1 sono mostrate le medie calcolate. La media stimata è indicata in linea rossa, la media ottenuta mediante averaging convenzionale in linea blu.

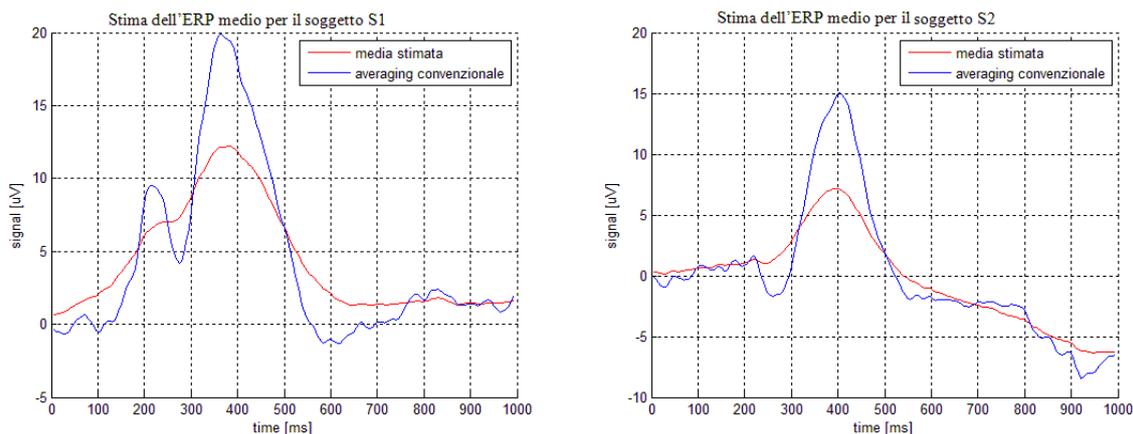


Figura 6.2.1: Esempi di ERP medio di due soggetti sani, S1 (sx) ed S2 (dx). In rosso la curva media stimata ed in blu la media ottenuta mediante averaging convenzionale

Si noti che la media stimata è inferiore in ampiezza alla media ottenuta mediante averaging convenzionale. Questo aspetto è correlato all'effetto di sottostima della componente ERP analizzato in precedenza. In figura 6.2.2 sono mostrati esempi di stima single trial per il soggetto S1. La sweep registrata è indicata in linea magenta, la stima ERP in linea blu ed infine la media stimata in linea rossa.

Infine, in figura 6.2.3 sono mostrati esempi di stima B2S relativi al soggetto S2.

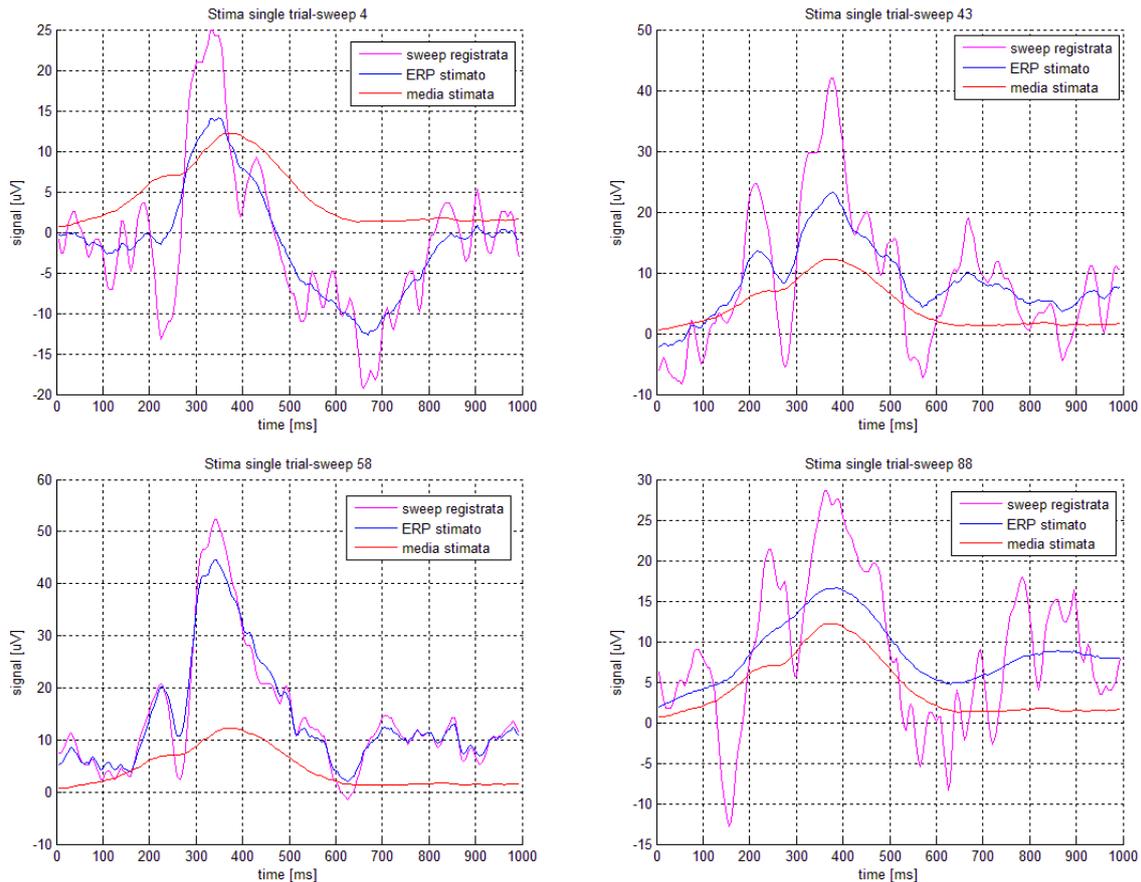


Figura 6.2.2: Esempi di stime single trial ottenute dal metodo B2S per il soggetto S1. In magenta la sweep registrata, in rosso la media stimata ed in blu la stima B2S.

In questa analisi, non è possibile definire la correttezza della stima della componente ERP. Infatti, a differenza del caso simulato, non si è a conoscenza della vera componente P300. I commenti sono così relativi all'individuazione di una deflessione positiva circa a 300 ms e questa caratteristica si evidenzia negli esempi proposti. Per esempio, nella sweep 4 del soggetto S1, il picco significativo si presenta a circa 350 ms. Per il soggetto S2, il grafico relativo alla sweep 22 mostra una latenza poco superiore ai 400 ms.

In figura 6.2.4 sono mostrate le stime dell'ERP medio per i due soggetti ALS, a sinistra il soggetto M3 e a destra il soggetto M5.

6. APPLICAZIONE DELLE TECNICHE B2S E MTL PER LA STIMA DELLA P300 DA DATI REALI

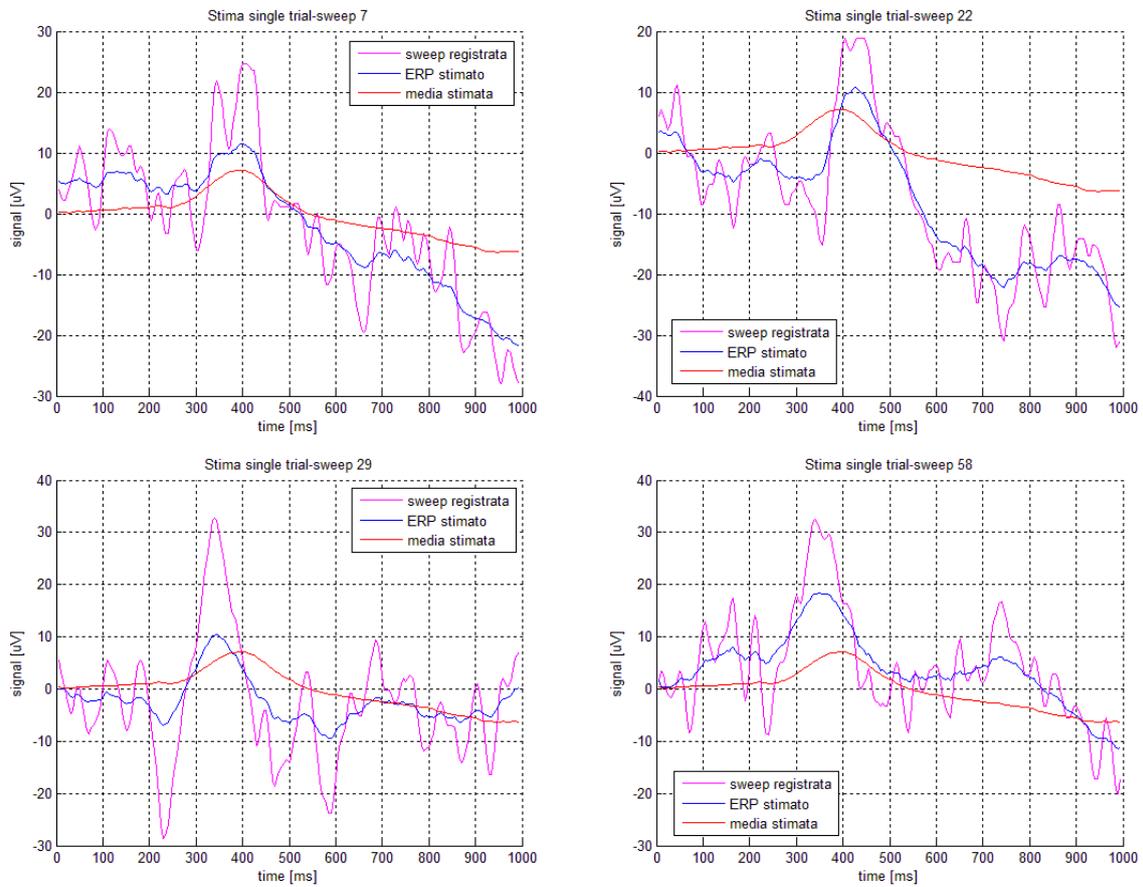


Figura 6.2.3: Esempi di stime single trial per il soggetto S2. In magenta la sweep registrata, in rosso la media convenzionale ed in blu la stima B2S.

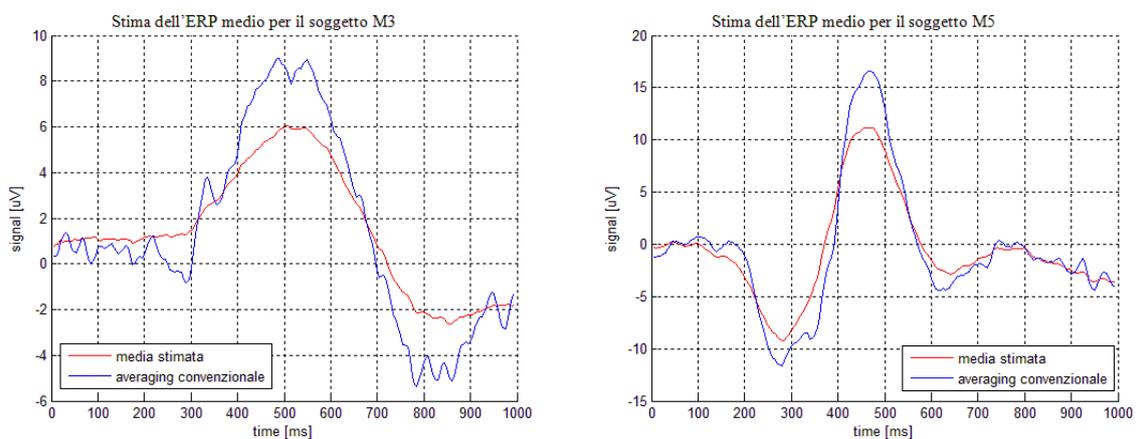


Figura 6.2.4: Esempi di ERP medio dei soggetti ALS, in blu la media ottenuta mediante averaging convenzionale, in rosso la media stimata mediante B2S

E' evidente come la media stimata si avvicini abbastanza, in termini di andamento, a quella ottenuta mediante averaging convenzionale. In entrambi i casi però la stima è sottostimata rispetto alla media ottenuta mediante averaging convenzionale. I risultati della stima single-trial per i soggetti M3 e M5 sono riportati rispettivamente in figura 6.2.5 e 6.2.6. La legenda è la medesima. La sweep registrata è indicata in linea magenta, la stima single tria in linea blu e la media stimata in linea rossa.

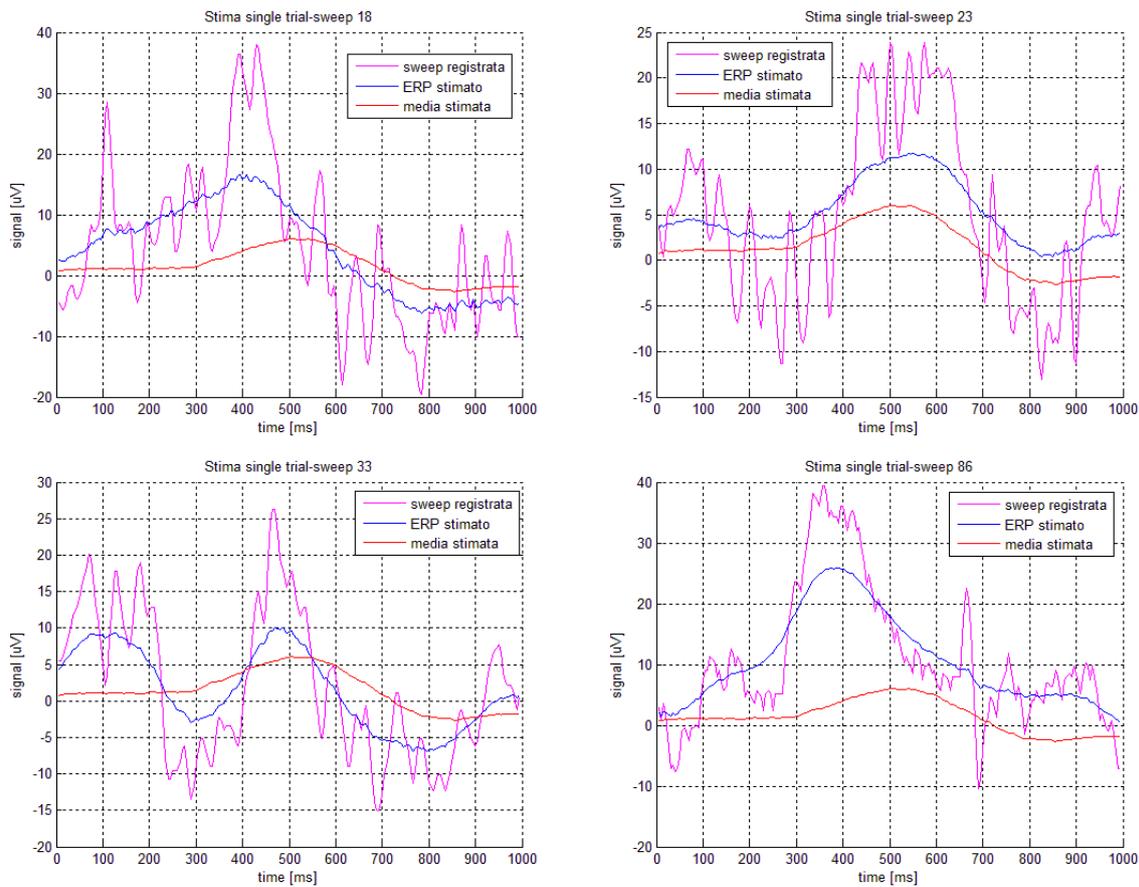


Figura 6.2.5: Esempi di stime single trial per il soggetto ALS M3. In magenta la sweep registrata, in rosso la media stimata ed in blu la stima ERP mediante il metodo B2S.

In questo caso, l'algoritmo riesce a fornire dei profili non troppo smooth per la componente P300. In particolare, anche in questo caso è possibile evidenziare la stima del potenziale ERP valutando l'occorrenza di una deflessione positiva a circa 300 ms. Entrambi i soggetti ALS presentano valori di latenza della P300 a circa 400 ms (si noti la stima single trial della sweep 86 per il soggetto M3 e la sweep 13 per il soggetto M5).

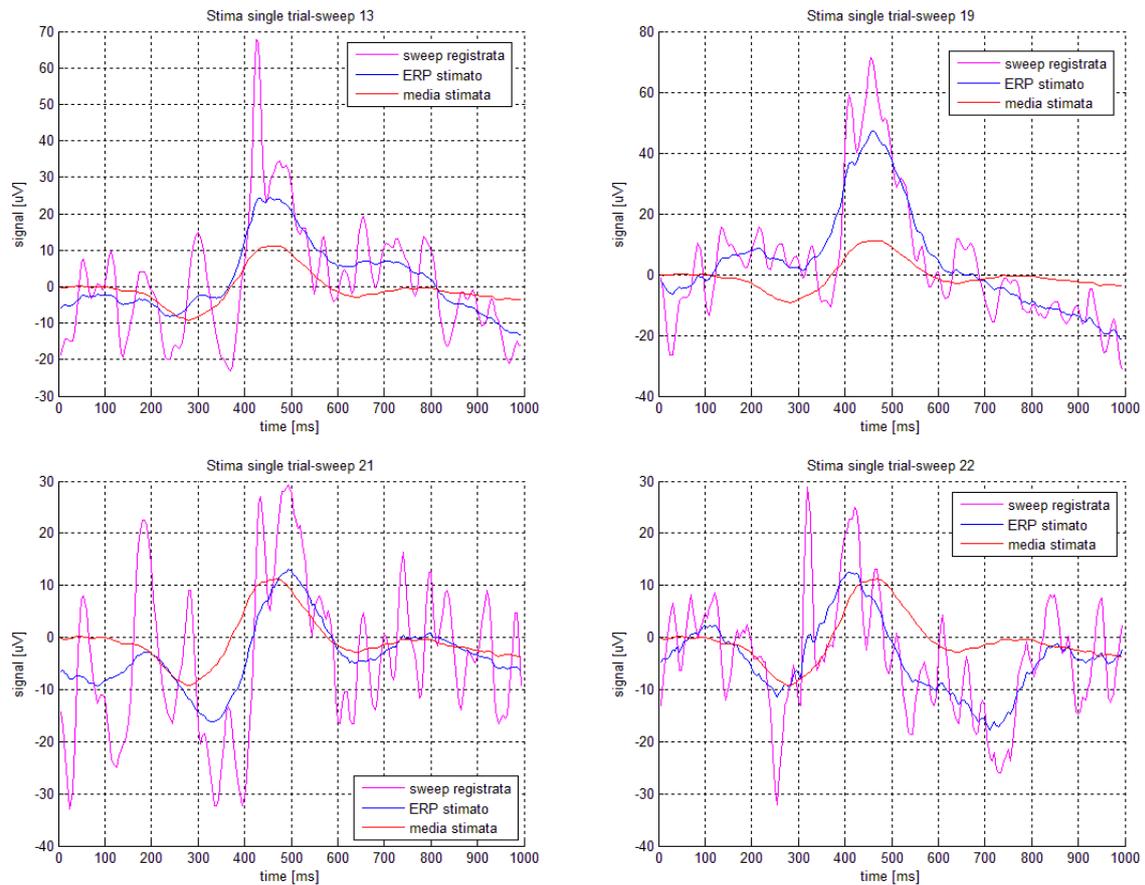


Figura 6.2.6: Esempi di stime single trial per il soggetto ALS M5. In magenta la sweep registrata, in rosso la media stimata ed in blu la stima ERP mediante il metodo B2S.

6.2.1 Casi particolari

Ispezionando le diverse stime B2S ottenute, si è notato che il caso di oversmoothing è ancora presente ed in modo abbastanza marcato. Inoltre, tale problematica non è relativa ad un solo soggetto, ma è un fenomeno che riguarda un certo numero di sweeps di tutti i soggetti. Ad esempio, il soggetto S1 ed il soggetto M3, per i quali erano state ottenute buone stime, presentano evidenti casi di oversmoothing. In figura 6.2.7 sono riportati alcuni esempi. Il pannello di sinistra fa riferimento al soggetto di controllo S1, mentre nel pannello di destra il soggetto ALS M3. La sweep registrata è indicata in linea magenta, la stima single tria in linea blu e la media stimata in linea rossa.

La stima ottenuta è molto piatta e non si riesce ad individuare alcun picco che possa somigliare alla componente P300. Ciò accade in ambedue i casi è a conferma del fatto che questo fenomeno non dipende dalla patologia del soggetto. Questo effetto è ancora legato al prior fornito per il calcolo del vettore di riferimento, in modo analogo a quanto visto per i dati simulati. Infatti, nel capitolo 8 verranno mostrate delle stime a cui è stato modificato il prior per il calcolo del vettore μ ed i miglioramenti sono evidenti.

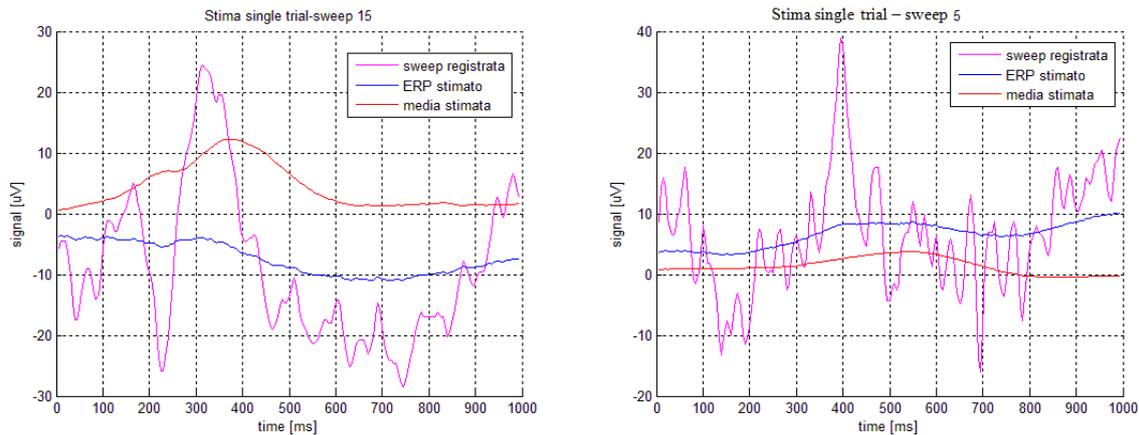


Figura 6.2.7: Esempi di stime single tria mediante B2S affette da oversmoothing. A sinistra la stima del soggetto sano S1, a destra la stima del paziente ALS M3.

6.3 Metodo MTL: risultati

Analogamente a quanto visto nel paragrafo dedicato alle stime B2S, si procede con quelle relative al metodo MTL, considerando i risultati ottenuti dalle sweeps target degli stessi soggetti.

Si inizia dall'analisi delle stime per i soggetti sani S1 e S2. In figura 6.3.1 si visualizzano gli andamenti ERP medi. La curva di popolazione stimata è indicata in linea rossa, mentre la media ottenuta dall'averaging convenzionale è in linea blu.

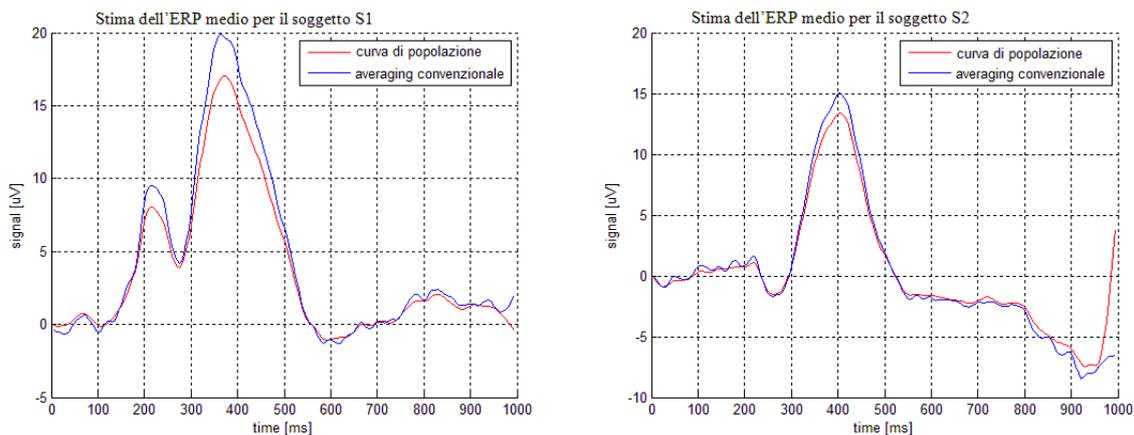


Figura 6.3.1: Esempi di ERP medio dei soggetti sani. In blu la media ottenuta mediante averaging convenzionale, in rosso la curva di popolazione MTL.

La curva di popolazione è molto vicina a quella ottenuta dall'averaging convenzionale. Tale risultato è evidente sia per il soggetto S1 che per il soggetto S2. Questo effetto è probabilmente legato alla tipologia di stima fornita dall'algoritmo MTL. La stima single

trial segue l'andamento della sweep registrata, eliminando il rumore localmente. Si mostrano alcuni esempi esplicativi di stima single trial per il soggetto S1 ed S2 rispettivamente in figura 6.3.2 e 6.3.3. La sweep registrata è indicata in linea magenta, la stima single trial ottenuta dal metodo MTL in linea blu ed infine la curva di popolazione in linea rossa.

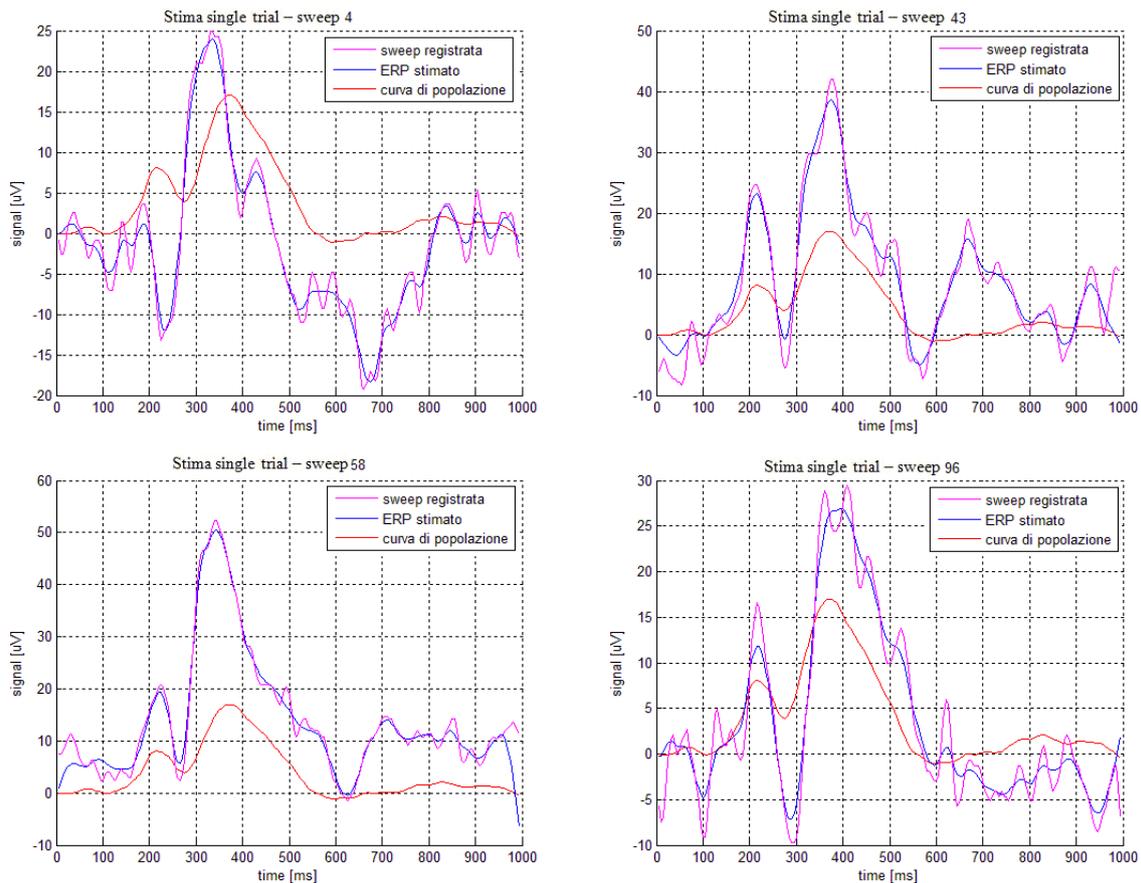


Figura 6.3.2: Esempi di stime single trial per il soggetto sano S1. In magenta la sweep registrata, in rosso la curva di popolazione ed in blu la stima ottenuta mediante MTL.

In questo caso le stime MTL sono buone e ben distinguibili dalla curva di popolazione. Si nota che in alcune sweeps, la stima è molto aderente al segnale grezzo. Questo effetto potrebbe essere legato al fatto che le sweeps sono poco rumorose e quindi la parte che viene eliminata risulta ridotta rispetto ad una sweep con maggiori oscillazioni. Medesime osservazioni per i grafici del soggetto sano S2. Anche in questo caso le sweeps sono poco rumorose e quindi la stima MTL è abbastanza buona. Essa si distingue dalla curva di popolazione, segue i dati eliminando comunque le piccole oscillazioni.

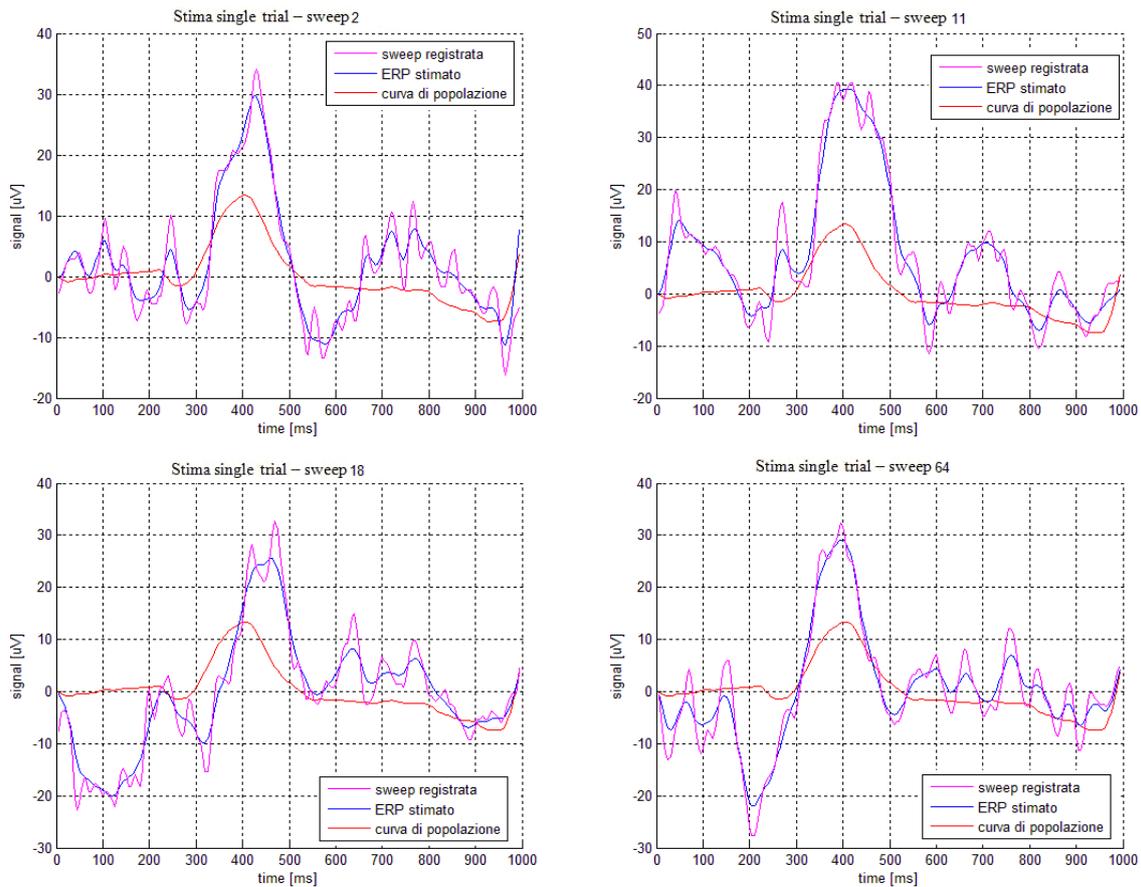


Figura 6.3.3: Esempi di stime single trial per il soggetto sano S2. In magenta la sweep registrata, in rosso la curva di popolazione ed in blu la stima ERP ottenuta mediante MTL.

Si prosegue l'analisi introducendo le stime ottenute per i soggetti ALS M3 ed M5. In figura 6.3.4 sono riportati i profili ERP medi. La curva di popolazione stimata dall'algoritmo MTL è indicata in linea rossa, quella ottenuta mediante averaging convenzionale in linea blu.

In entrambi gli esempi proposti, la curva di popolazione fornisce una stima del profilo ERP medio simile a quello dell'averaging convenzionale. In particolare, a differenza di quest'ultimo, la curva stimata è più regolare e non contiene oscillazioni spurie.

Si prosegue l'analisi mostrando i risultati per i soggetti ALS M3 ed M5 rispettivamente in figura 6.3.5 e 6.3.6. La legenda è la medesima, con la sweep registrata in linea magenta, la stima single trial in linea blu e la curva di popolazione in linea rossa.

6. APPLICAZIONE DELLE TECNICHE B2S E MTL PER LA STIMA DELLA P300 DA DATI REALI

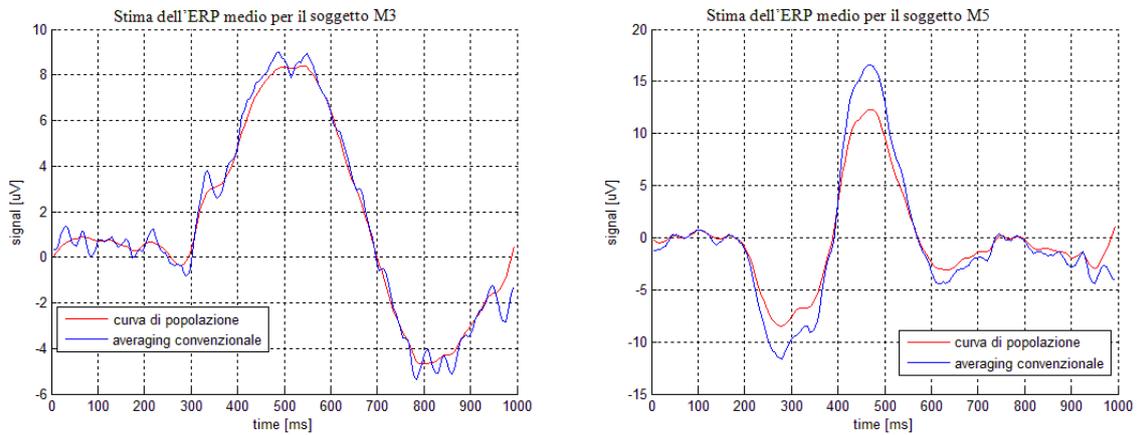


Figura 6.3.4: Esempi di ERP medio per i soggetti ALS. In blu la media ottenuta mediante averaging convenzionale, in rosso la curva di popolazione MTL.

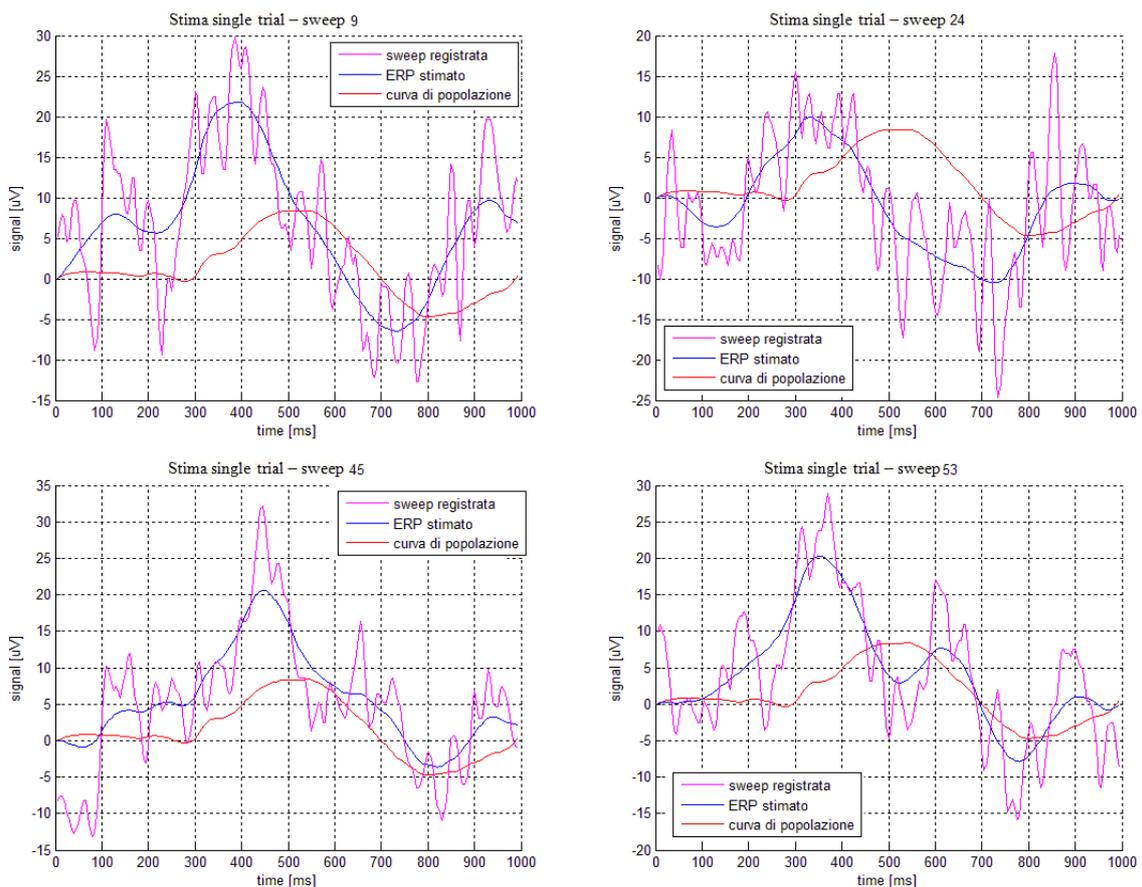


Figura 6.3.5: Esempi di stime single trial per il soggetto ALS M3. In magenta la sweep registrata, in rosso la curva di popolazione ed in blu la stima MTL.

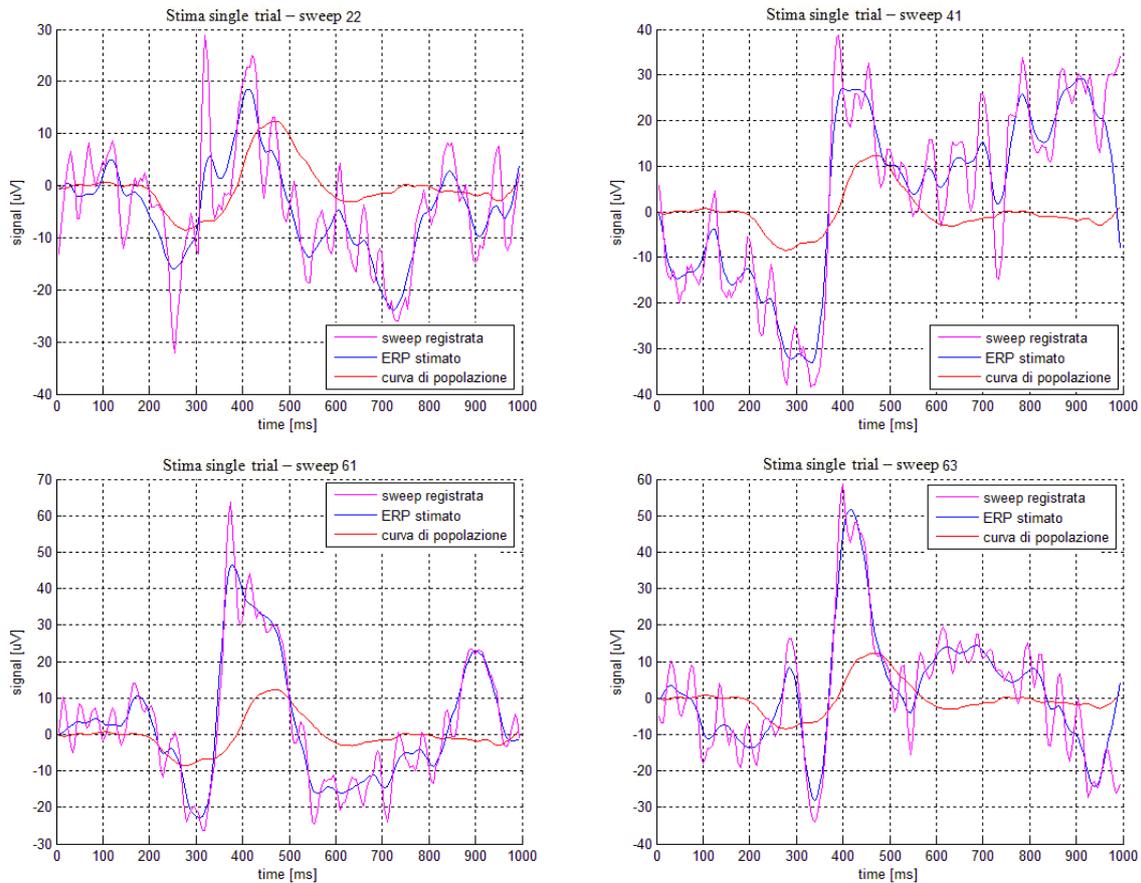


Figura 6.3.6: Esempi di stime single trial per il soggetto ALS M5. In magenta la sweep registrata, in rosso la curva di popolazione ed in blu la stima MTL.

Si noti come le stime single trial ottenute tramite il metodo MTL si discostano dalla curva di popolazione. Le stime sono regolari ed eliminano parte delle oscillazioni proprie del segnale grezzo. In particolare, negli esempi proposti è possibile individuare una deflessione positiva a circa 400 ms (ad esempio si consideri la sweep 45 per il soggetto M3 e la sweep 63 per il soggetto M5).

6.3.1 Casi particolari

Anche se le stime fornite dal metodo MTL sembrano in generale buone, può accadere che l'algoritmo stimi una curva di popolazione nulla. E' il caso del soggetto M1 la cui stima dell'ERP medio è riportata in figura 6.3.7. La media ottenuta mediante averaging convenzionale è indicata in linea blu, la curva di popolazione in linea rossa.

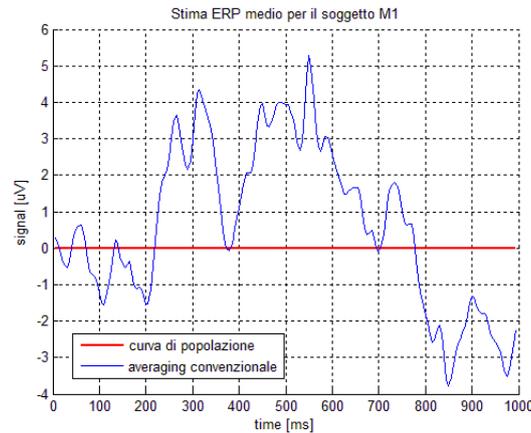


Figura 6.3.7: *Profilo ERP medio del soggetto M1. In rosso è rappresentata la curva di popolazione nulla, in blu la media ottenuta mediante averaging convenzionale.*

Ricordando la formula del modello dati per il segnale incognito e particolarizzandola per $\bar{z}(t) = 0$, si ottiene:

$$z_j(t) = \bar{z}(t) + \tilde{z}_j(t) \rightarrow z_j(t) = \tilde{z}_j(t)$$

Visto che la $\bar{z}(t)$ è indipendente dalla singola sweeps e rappresenta invece una qualche caratteristica comune a tutta la popolazione, il risultato sopra ottenuto indica che ogni sweep appartenente alla popolazione omogenea del soggetto M1 è indipendente l'una dall'altro. Questo fenomeno, infatti, non vincola le stime MTL delle singole sweeps, stime che peraltro sono abbastanza buone e sono riportate in figura 6.3.8.

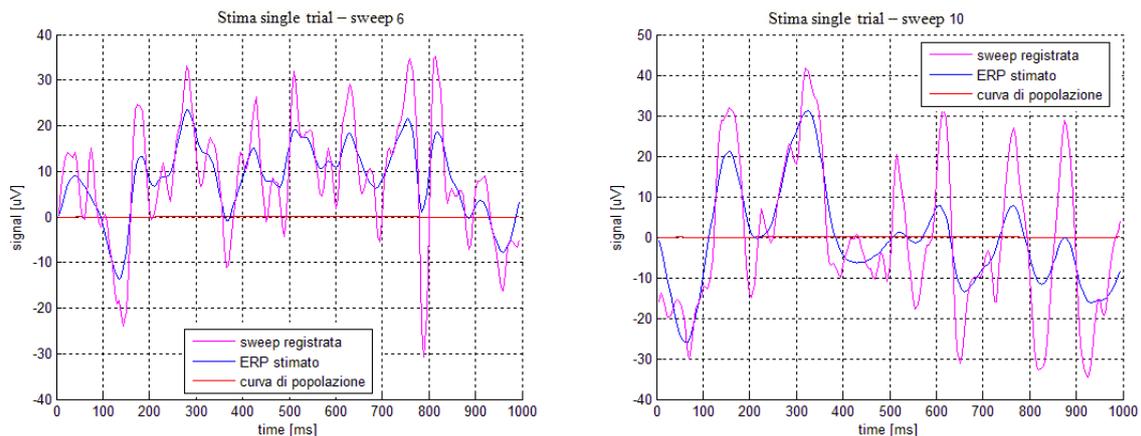


Figura 6.3.8: *Esempi di stime single trial ottenute dal metodo MTL per il soggetto M1.*

Dopo aver introdotto le stime single trial ottenute con il metodo MTL, è possibile fornire un confronto con le stime single trial ottenute con il metodo B2S in caso di over-smoothing. Si precisa così che l'over-smoothing è sicuramente legato al processo di stima, nonché di assunzioni del modello del segnale incognito scelto, piuttosto che alla presenza od assenza della componente P300 (tra l'altro si ricorda che le stime sotto ottenute da sweeps target, ovvero tracce in cui almeno in teoria deve essere presente la componente ERP). Si propone in figura 6.3.9 il confronto di una stima single trial ottenuta dal metodo B2S e MTL rispetto alla medesima sweep del soggetto M3.

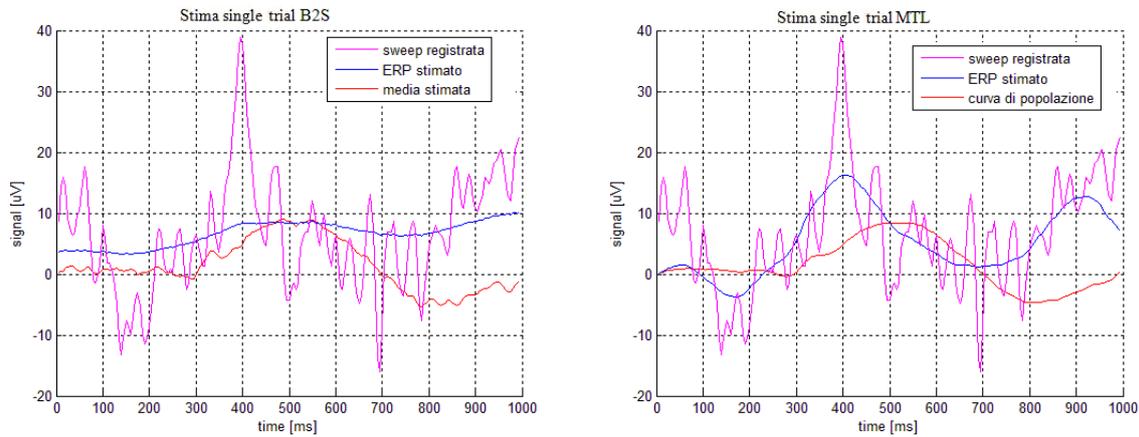


Figura 6.3.9: *Confronto della stima single trial con il metodo B2S ed il metodo MTL della stessa sweep del soggetto M3.*

Si noti come la stima single trial fornita dal metodo MTL individua una deflessione positiva a circa 400 ms. La stima single trial ottenuta dal metodo B2S è, invece, piatta.

6.4 Conclusioni

In questo capitolo sono stati considerati i risultati ottenuti in seguito all'applicazione delle due tecniche a dati reali. La difficoltà nell'interpretazione è evidente a causa della mancata conoscenza del segnale vero. Per entrambi i metodi sono state riportate le stime relative sia a soggetti ALS che a soggetti sani. Questo ha permesso di apprezzare la variabilità intersoggettiva sia dei dati acquisiti nonché delle stime ottenute. Non si sono rilevate invece delle differenze nette nei profili stimati fra soggetti sani e pazienti ALS.

Le stime B2S e MTL sono profondamente diverse l'una rispetto l'altra. La stima B2S fornisce stime più regolari. Il metodo MTL, invece, fornisce dei profili stimati che, da un lato, ripropongono l'andamento dei dati grezzi dall'altro si presentano depurate dalle oscillazioni presenti sulla sweep registrata.

Non è semplice stabilire quale delle due metodiche fornisce le stime migliori: la stima MTL sicuramente sembra quella più realistica, ma a volte tende a rimanere legata all'andamento proprio della sweep grezza.

A questo punto bisogna ricordare lo scopo della tesi: valutare la possibilità di inserire uno step di pre-processing basato su questi algoritmi all'interno di un sistema BCI. Questo implica che poco importa quale visivamente sia la stima migliore.

Il confronto e quindi la valutazione della migliore o peggiore tecnica deve essere valutata in termini di errore da parte del classificatore addestrato rispetto le tracce ottenute dalle stime B2S e MTL.

Questo aspetto cruciale sarà oggetto del capitolo successivo.

Capitolo 7

Valutazione off-line del sistema BCI con tecniche di pre-processing bayesiane

7.1 Schema di addestramento del classificatore

Come descritto nel capitolo 1, il classificatore utilizzato all'interno del sistema BCI viene addestrato sulla base delle features estratte da segnali opportunamente elaborati. In particolare, in questo capitolo, si valuterà la possibilità di fornire in ingresso al classificatore delle features calcolate sulle stime fornite dai due metodi illustrati nei capitoli precedenti. Lo schema di addestramento del classificatore è il medesimo ed è indicato in figura 7.1.1.

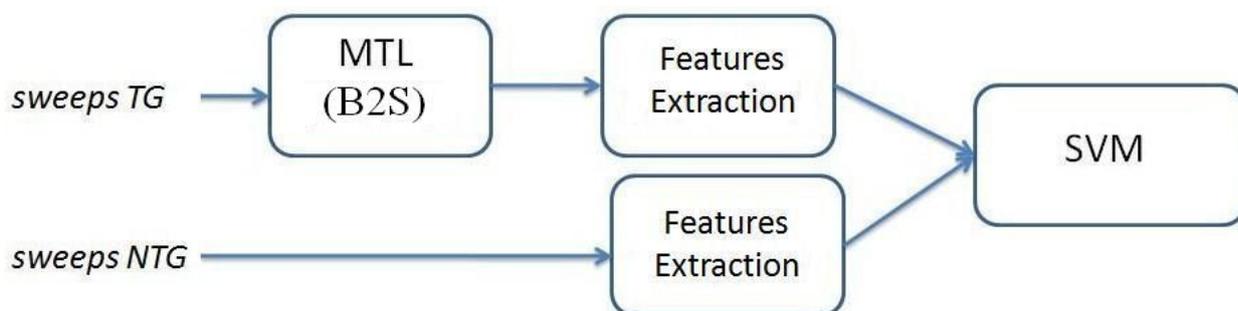


Figura 7.1.1: *Schema di addestramento del classificatore SVM. Solo le sweeps target vengono elaborate dal metodo MTL(o dal metodo B2S). Le features estratte sono i 78 parametri sulla base dei quali il classificatore SVM verrà addestrato.*

Tale schema verrà applicato separatamente per le due tecniche e solo alle sweeps target. Questa scelta è dovuta alla natura delle tecniche single trial utilizzate. Sia il metodo B2S che il metodo MTL sono tecniche che stimano la componente P300 da una generica sweep passatagli come argomento. Per cui non valutano l'eventuale presenza o no della componente ERP, ma in base all'informazione che ricavano dal segnale ne forniscono una stima. Può accadere che, in determinate situazioni, una componente particolare di rumore possa

essere scambiata per una P300. Inoltre, esistono casi in cui le sweeps non target possono presentare una sorta di componente ERP, la quale verrebbe stimata dall'algoritmo.

Il problema è più evidente in termini di classificatore: se infatti vengono fornite come estratte da classi diverse delle features estratte da stime comparabili tra loro, è chiaro che il classificatore non sarà in grado di discriminare nettamente le due classi e commetterà degli errori elevati.

Bisogna sottolineare che questi metodi vengono applicati all'interno di un sistema BCI per la prima volta, per cui queste sono le prime prove condotte in questo senso. In particolare, con uno schema di applicazione di figura 7.1.1 ci si concentra sulla capacità del classificatore SVM di riconoscere la componente P300.

L'iter che verrà seguito è il seguente: a partire dal numero totale di sweeps target stimate e non target grezze, si creano due insiemi di dati. Il primo è quello più numeroso e viene definito *learning set*. Le sweeps all'interno di questo gruppo verranno direttamente utilizzate per l'addestramento del classificatore. Il secondo gruppo, meno numeroso, è il *testing set*, dataset costituito sia da sweeps target che non target che non viene utilizzato all'interno del processo di addestramento. Lo scopo di questo gruppo è testare su nuove sweeps, mai utilizzate per l'apprendimento, il classificatore SVM creato precedentemente. Queste andranno a simulare un test reale, in cui il classificatore non sa a priori la classe di appartenenza della sweep.

7.2 Valutazione delle performance del sistema BCI

Per la valutazione degli errori commessi dal classificatore SVM, il programma implementato nell'istituto IRCCS del San Camillo prevede l'analisi di tre indici fondamentali per ogni classificatore creato.

Prima di introdurli, si definiscono i seguenti parametri

1. $n_1 \rightarrow$ numero totale di stimoli non target.
2. $n_2 \rightarrow$ numero totale di stimoli target.
3. $VP \rightarrow$ epoche target classificate correttamente.
4. $VN \rightarrow$ epoche non target classificate correttamente.

Gli indici sono i seguenti

- ERRORE SUL RICONOSCIMENTO DELLA P300 ($ep3$): rappresenta il complementare del rapporto fra quante tracce target sono state classificate correttamente rispetto al numero totale di stimoli target.

In formule

$$ep3 = 1 - \frac{VP}{n_2} \quad (7.2.1)$$

- ERRORE SUL RICONOSCIMENTO DELLA NON P300 ($enp3$): rappresenta il complementare del rapporto fra quante tracce non target sono state classificate correttamente rispetto al numero totale di stimoli standard. In formule

$$enp3 = 1 - \frac{VN}{n_1} \quad (7.2.2)$$

- ERRORE SUL RICONOSCIMENTO ($etot$): rappresenta il complementare del rapporto tra quante tracce, sia target che non target, sono state riconosciute correttamente rispetto al numero complessivo di stimoli, sia target che standard. In formule

$$etot = 1 - \frac{VP + VN}{n_1 + n_2} \quad (7.2.3)$$

Infine, si definisce PERFORMANCE del classificatore la quantità $1 - etot$.

La procedura prevede così che per ogni classificatore addestrato, il calcolo dei tre parametri ($ep3$, $enp3$, $etot$) sia rispetto al learning set che per il testing set. Chiaramente, ai fini dell'analisi interessa capire come il classificatore si comporta quando non ha conoscenza sulle sweeps in ingresso, per cui si valuteranno i tre indici sul testing set.

Affinché si potesse valutare la variabilità di tali parametri, sono stati addestrati un certo numero di SVM e per ogni classificatore, il learning e testing set vengono costruiti in modo random. Questa procedura è applicata sia ai dati simulati che ai dati reali, in modo tale da poter valutare infine le performance medie. Proprio per questo, nelle tabelle che si visualizzeranno nel paragrafo successivo, ad ogni valore medio di $ep3$, $enp3$ ed $etot$ del testing set verrà associata la standard deviation, in modo da valutare la variabilità dell'errore rispetto a tutti i classificatori creati.

Inoltre, per completare l'analisi, verrà anche fornito un nuovo indice di valutazione: l'errore standard (ES), una misura relativa alla stima della deviazione standard dello stimatore, la quale fornisce una misura della sua precisione. In formule

$$ES = \frac{std}{\sqrt{N}}$$

in cui std è la standard deviation relativa ad uno specifico errore, N è il numero di classificatori totali addestrati. Chiaramente, verrà definito separatamente l'errore ES per i tre errori $ep3$, $enp3$ ed $etot$.

7.3 Risultati dai dati simulati

Si descrive brevemente la procedura seguita per la costruzione del learning e del testing set. Il numero di sweeps target impiegate risulta pari ad 80. Per completare tale dataset, bisogna inserire le sweeps non target simulate e sceglierne il numero adeguato. A questo proposito si valuti il paradigma descritto in figura 2.1.1 del capitolo 2; esso prevede che per ogni stimolo target vi siano tre stimoli non target. Quindi, per ricostruire nel dataset simulato una numerosità reale, si è deciso di mantenere la stessa proporzione 1:3 tra sweeps target e sweeps non target. Per cui se 80 è il numero complessivo di sweeps target, il numero totale di sweeps non target sarà pari a 240 sweeps.

Queste sweeps sono state suddivise tra learning e testing set come riportato in tabella 7.1.

	nTG	nNTG	nTOT
Learning	55	165	220
Testing	25	75	100

Tabella 7.1: *Numerosità del learning e testing set per dati simulati.*

Volutamente si è scelta una numerosità non eccessiva per mimare il caso reale. Questa suddivisione in learning e testing set è uguale sia per l'applicazione dello schema 7.1.1 con B2S che con MTL. Il numero totale di classificatori creati è pari a 40 e per ogni SVM, il learning e testing set sarà diverso perché le sweeps per i due dataset vengono selezionate in modo random dall'insieme complessivo (e quindi per le target e non target), rispettando però la numerosità riportata in tabella 7.1.

Gli errori medi sono riportati nella tabella 7.2.

TESTING	B2S	MTL	p-value
ep3 (std)	0.0126 (0.0299)	0.0183 (0.0259)	0.142*
enp3 (std)	0.0080 (0.0097)	0.0179 (0.0105)	$p < 10^{-3}$ *
etot (std)	0.0091 (0.0108)	0.0180 (0.0082)	$p < 10^{-4}$ *
Errore Standard			
ep3	0.0051	0.0044	
enp3	0.0016	0.0018	
etot	0.0018	0.0014	

Tabella 7.2: *Errori medi sui dati simulati.*

Si nota subito che l'errore è molto basso per i tre indici. Si ricorda che ciò a cui si presterà maggiore attenzione è l'errore sul riconoscimento della P300. Per dare un significato statistico alle differenze di errori fra i due metodi, è stato applicato il test Mann-Whitney per campioni indipendenti. Ciò che si nota è che l'errore medio sul riconoscimento della P300 è del tutto confrontabile fra i due metodi, per cui, almeno sui dati simulati, non c'è una metodica che prevale sull'altra.

7.4 Risultati dai dati reali

In questo paragrafo vengono presentati i risultati relativi alla performance del classificatore addestrato su dati reali. In particolare, alle sweeps acquisite, sia target che non target, viene applicata la rimozione dell'artefatto mediante sogliatura, ottenendo le sweeps *ssa2* ed *ssa1*, la cui numerosità complessiva è indicata in tabella 6.1. Elaborate le sweeps target, si costruisce per ogni soggetto il learning e testing set, la cui numerosità sarà dipendente dal numero di sweeps registrate per soggetto. Per questo, il learning e testing set sono stati costruiti ad hoc per ogni soggetto con numerosità indicata nella seguente tabella 7.3. Le colonne indicano le seguenti quantità

- L_TG → numero di sweeps target per il learning set;
- L_NTG → numero di sweeps non target per il learning set;
- T_TG → numero di sweeps target per il testing set;
- T_NTG → numero di sweeps non target per il testing set;
- $totL$ → numero totale di sweeps sia target che non target per il learning set;
- $totT$ → numero totale di sweeps sia target che non target per il testing set.

sbj	L_TG	L_NTG	T_TG	T_NTG	totL	totT
M1	150	450	50	147	600	197
M2	80	240	55	137	320	192
M3	90	200	26	64	290	90
M4	80	149	30	60	229	90
M5	120	360	49	231	480	280
S1	90	250	28	77	340	105
S2	140	420	54	107	560	161
S3	150	450	50	96	600	146

Tabella 7.3: *Numerosità del Learning e Testing set per ogni soggetto.*

Il numero di classificatori SVM addestrati è pari a 40 e per ogni fase di addestramento, il learning test ed il testing set viene creato in modo random rispettando la numerosità indicata in tabella 7.3. I risultati relativi agli errori medi ottenuti sui diversi soggetti sono rappresentati nelle tabelle seguenti. In ciascuna tabella viene riportato il valore di significatività ottenuto con il test di Wilcoxon per campioni appaiati (variante paired), in modo da confrontare gli errori ottenuti con le due tecniche. Inoltre, accanto a ciascuna tabella, sono anche visualizzati i boxplot relativi alle distribuzioni degli errori sul riconoscimento della P300 (ep3), in modo da interpretare il risultato del test statistico anche attraverso la distribuzione dei valori forniti dalle due metodologie.

7. VALUTAZIONE OFF-LINE DEL SISTEMA BCI CON TECNICHE DI PRE-PROCESSING BAYESIANE

TESTING	B2S	MTL	p-value
ep3 (std)	0.0550 (0.0285)	0.0775 (0.0329)	$p < 10^{-2}$ *
enp3 (std)	0.0012 (0.0030)	0.0333 (0.0159)	$p < 10^{-7}$ *
etot (std)	0.0148 (0.0075)	0.0445 (0.0143)	$p < 10^{-7}$ *

Errore Standard		
ep3	0.0045	0.0052
enp3	0.0005	0.0025
etot	0.0012	0.0023

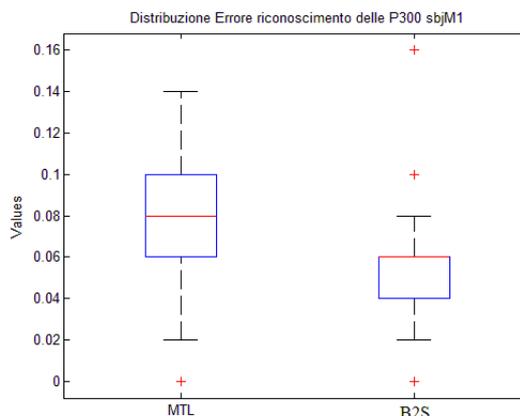


Tabella 7.4: Tabella relativa al soggetto ALS *M1*.

Per il soggetto M1, i tre indici di errore assumono valori molto bassi. In questo caso, la tecnica B2S ha un errore ep3 più basso del MTL, differenza significativa testimoniata anche dal p-value corrispondente. Anche il grafico dei boxplot conferma ciò: la mediana B2S è più bassa e la sua distribuzione meno ampia, anche se comunque questa presenta degli outliers.

TESTING	B2S	MTL	p-value
ep3 (std)	0.0373 (0.0211)	0.0059 (0.0094)	$p < 10^{-6}$ *
enp3 (std)	0.0077 (0.0065)	0.0038 (0.0049)	$p < 10^{-2}$ *
etot (std)	0.0161 (0.0065)	0.0044 (0.0044)	$p < 10^{-6}$ *

Errore Standard		
ep3	0.0033	0.0015
enp3	0.0010	0.0008
etot	0.0010	0.0007

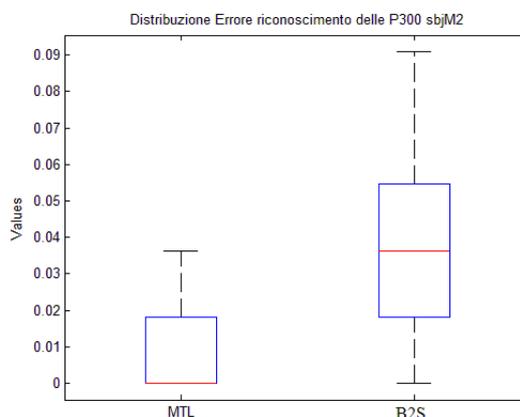
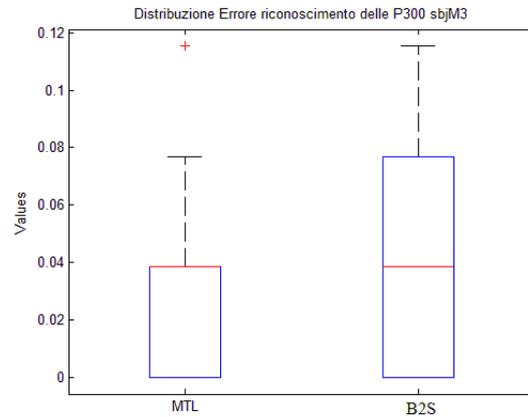


Tabella 7.5: Tabella relativa al soggetto ALS *M2*.

Per il soggetto M2, gli errori ep3, enp3 ed etot sono molto bassi, ma si nota il netto miglioramento nell'errore ep3 con la tecnica MTL rispetto a quella B2S, significativa differenza confermata dal test statistico. Anche nel grafico boxplot si individua la netta differenza della distribuzione, a favore del caso MTL perché più compatta e con mediana più bassa.

TESTING	B2S	MTL	p-value
ep3 (std)	0.0375 (0.0360)	0.0327 (0.0317)	0.512
enp3 (std)	0.0211 (0.0173)	0.0293 (0.0185)	$p < 10^{-2}$ *
etot (std)	0.0258 (0.0173)	0.0303 (0.0157)	0.176

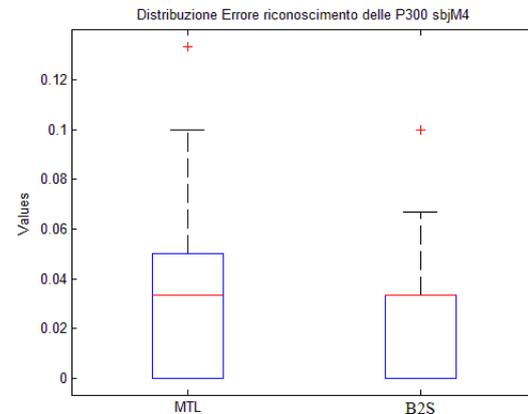
Errore Standard			
ep3	0.0057	0.0050	
enp3	0.0027	0.0029	
etot	0.0027	0.0025	

Figura 7.4.1: Tabella relativa al soggetto ALS **M3**.

Per il soggetto M3, i tre errori ep3, enp3 ed etot sono molto bassi ed in questo caso non si evidenziano delle grandi differenze nell'errore ep3 fra B2S ed MTL. Questo è confermato dal test statistico, che fornisce un p-value abbastanza elevato. Nel grafico boxplot si nota che le mediane hanno uguale valore e che le distribuzioni, in generale, sono effettivamente molto simili.

TESTING	B2S	MTL	p-value
ep3 (std)	0.0267 (0.0309)	0.0350 (0.0333)	0.290
enp3 (std)	0.0392 (0.0265)	0.0292 (0.0196)	0.0834
etot (std)	0.0350 (0.0173)	0.0311 (0.0196)	0.350

Errore Standard			
ep3	0.0049	0.0053	
enp3	0.0042	0.0031	
etot	0.0027	0.0031	

Tabella 7.6: Tabella relativa al soggetto ALS **M4**.

Per il soggetto M4, gli errori ep3, enp3 ed etot sono molto bassi e l'errore ep3 ha un valore confrontabile fra le due tecniche. Questo è confermato da un p-value abbastanza alto. Anche il grafico boxplot conferma ciò: la mediana è la stessa e le due distribuzioni sono abbastanza simili.

7. VALUTAZIONE OFF-LINE DEL SISTEMA BCI CON TECNICHE DI PRE-PROCESSING BAYESIANE

TESTING	B2S	MTL	p-value
ep3 (std)	0.0622 (0.0362)	0.1010 (0.0540)	$p < 10^{-3}$ *
enp3 (std)	0.0286 (0.0128)	0.0445 (0.0164)	$p < 10^{-4}$ *
etot (std)	0.0345 (0.0118)	0.0544 (0.0139)	$p < 10^{-6}$ *
Errore Standard			
ep3	0.0057	0.0085	
enp3	0.0020	0.0026	
etot	0.0019	0.0022	

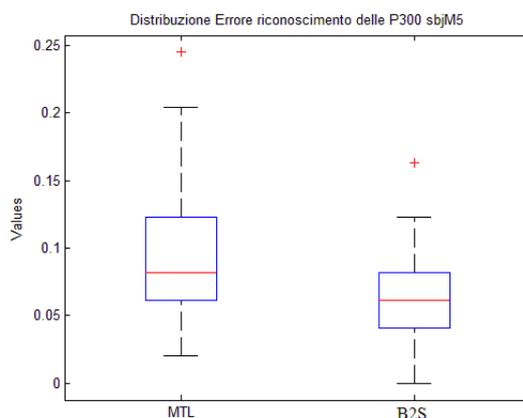


Tabella 7.7: *Tabella relativa al soggetto ALS M5.*

Per il soggetto M5, gli errori ep3, enp3 ed etot si mantengono bassi. Rispetto ad altri casi però aumenta leggermente l'errore ep3, che rimane più basso nel caso B2S rispetto al caso MTL. Questo è confermato dal test statistico che fornisce un valore del p-value molto basso. Lo stesso grafico boxplot è inequivocabile: la distribuzione B2S è più bassa e meno sparsa rispetto al caso MTL.

TESTING	B2S	MTL	p-value
ep3 (std)	0.0527 (0.0299)	0.0973 (0.0673)	$p < 10^{-3}$ *
enp3 (std)	0.0084 (0.0103)	0.0542 (0.0274)	$p < 10^{-7}$ *
etot (std)	0.0202 (0.0115)	0.0657 (0.0217)	$p < 10^{-7}$ *
Errore Standard			
ep3	0.0047	0.0106	
enp3	0.0016	0.0043	
etot	0.0018	0.0034	

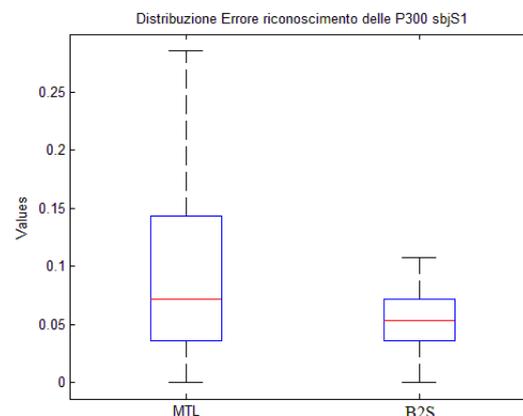
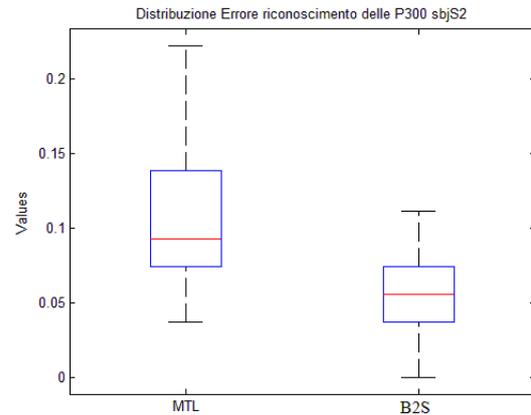


Tabella 7.8: *Tabella relativa al soggetto sano S1.*

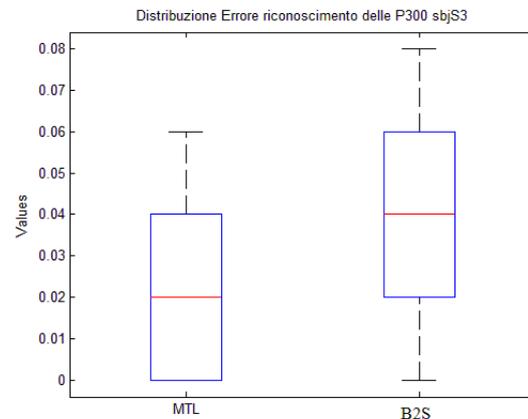
Per il soggetto S1, gli errori ep3, enp3 ed etot sono bassi, mentre l'errore ep3 è sicuramente più alto nel caso MTL. Ciò è confermato dal risultato del test statistico che conferma le significative differenze. Anche il grafico boxplot non lascia dubbi: la distribuzione B2S è meno sparsa e la sua mediana è inferiore.

TESTING	B2S	MTL	p-value
ep3 (std)	0.0556 (0.0265)	0.1037 (0.0426)	$p < 10^{-4}$ *
enp3 (std)	0.0273 (0.0183)	0.0343 (0.0186)	0.05*
etot (std)	0.0368 (0.0150)	0.0576 (0.0197)	$p < 10^{-4}$ *
Errore Standard			
ep3	0.0042	0.0067	
enp3	0.0029	0.0029	
etot	0.0024	0.0031	

Tabella 7.9: *Tabella relativa al soggetto sano S2.*

Per il soggetto S2, gli errori ep3, enp3 ed etot sono bassi, anche se l'ep3 è un po' più alto rispetto ad altri casi. Quindi il metodo B2S fornisce un valore ep3 più basso rispetto al caso MTL, risultato significativo perché confermato dal valore basso di p-value. Anche il grafico boxplot rappresenta la situazione in modo chiaro: la distribuzione degli errori per il metodo MTL è più ampia e con valori più alti rispetto al caso B2S.

TESTING	B2S	MTL	p-value
ep3 (std)	0.0350 (0.0231)	0.0195 (0.0176)	$p < 10^{-2}$ *
enp3 (std)	0.0057 (0.0081)	0.0227 (0.0156)	$p < 10^{-4}$ *
etot (std)	0.0158 (0.0086)	0.0216 (0.0134)	$p < 10^{-2}$ *
Errore Standard			
ep3	0.0037	0.0028	
enp3	0.0013	0.0025	
etot	0.0014	0.0021	

Tabella 7.10: *Tabella relativa al soggetto sano S3.*

Per il soggetto S3, gli errori ep3, enp3 ed etot si mantengono molto bassi ed in particolare l'errore ep3 è nettamente più basso per MTL rispetto che al metodo B2S, risultato confermato dal test statistico che fornisce un valore del p-value molto basso. Lo stesso boxplot propone una mediana più bassa nel caso MTL rispetto a B2S.

In generale si può affermare che i valori degli errori ottenuti con le due tecniche sono molto bassi per tutti i soggetti e che le performance dei due metodi sono confrontabili. In particolare

- l'errore ep_3 fornito dal metodo MTL è più basso rispetto a quello fornito dal metodo B2S nei soggetti M2 ed S3;
- l'errore ep_3 fornito dal metodo B2S è più basso rispetto a quello fornito dal metodo MTL nei soggetti M1, M5, S1 ed S2;
- l'errore ep_3 fornito dal metodo MTL e dal metodo B2S è confrontabile nei soggetti M3 ed M4.

Si vogliono ora confrontare i risultati forniti da queste due tecniche in termini di errori ep_3 con quelli del sistema attualmente in funzione nell'istituto IRCCS (metodica di pre-processing ICA). In particolare, per gli errori relativi alla procedura ICA sono stati utilizzati quelli precedentemente calcolati in [6]. Per tale motivo, il soggetto S3 non verrà utilizzato per il confronto perché è un nuovo elemento del dataset. In tabella 7.11 sono riportati gli errori ep_3 ottenuti per il resto dei soggetti impiegando le tre tecniche.

Soggetto	ep3-B2S	ep3-MTL	ep3-ICA
M1	0.051	0.064	0.82
M2	0.035	0.002	0.54
M3	0.046	0.017	0.44
M4	0.023	0.065	0.28
M5	0.063	0.086	0.33
S1	0.046	0.102	0.28
S2	0.056	0.100	0.42

Tabella 7.11: *Tabella degli errori ep_3 ottenuti sui soggetti con i metodi B2S, MTL ed ICA.*

L'errore ep_3 è decisamente più basso con le due nuove metodiche B2S ed MTL rispetto a quanto calcolato con la metodica di pre-processing ICA. Questi sono risultati molto importanti anche perché il vantaggio dell'eventuale utilizzo dei due nuovi metodi nello step di processing delle sweeps, in luogo dell'ICA, è davvero notevole.

7.5 Conclusioni

In questo capitolo è stata valutata la performance del classificatore SVM addestrato sulla base delle features estratte dalle stime fornite dai metodi B2S e MTL.

In questa prima fase di validazioni off-line l'obiettivo è valutare se l'errore sul riconoscimento della P300 viene ridotto qualora al segnale vengano applicate le tecniche di pre-processing B2S oppure MTL.

L'analisi condotta sui dati simulati mostra che l'errore sul riconoscimento della P300 è molto basso per entrambe le tecniche e che i risultati ottenuti con le due metodiche sono del tutto confrontabili.

Anche l'analisi su dati reali ha dato ottimi risultati. In particolare, gli errori medi calcolati rispetto le metodiche B2S ed MTL sono, in generale, inferiori al 15%, un risultato molto importante e decisamente positivo. Inoltre, nel confronto delle due tecniche B2S ed MTL non si è individuata una metodica che fornisse risultati in assoluto migliori. In alcuni soggetti la tecnica MTL è quella più performante, in altri invece il metodo B2S. Si può concludere che così entrambe le tecniche presentano performance, sul riconoscimento della P300, sicuramente ottimali.

L'ultimo confronto fatto fa riferimento all'errore ep_3 rispetto al sistema attualmente in uso al San Camillo: Independent Component Analysis. Per i dati relativi a quest'ultima metodologia è stato riportato quanto trovato nel lavoro [6]. Sembra proprio che non ci sia confronto fra le tre metodologie: le due nuove tecniche forniscono errori nettamente più bassi.

Questi risultati molto incoraggianti terminano il lavoro di tesi, nonché la prima fase della validazione off-line delle due metodologie proposte.

Capitolo 8

Conclusioni e sviluppi futuri

8.1 Bilancio e considerazioni riassuntive

Una BCI è un sistema che crea un canale alternativo per la comunicazione tra il cervello di un soggetto e il mondo esterno, senza coinvolgere muscoli e nervi. I sistemi più diffusi sono quelli basati sul segnale EEG, ed in particolare, quello analizzato in questa tesi è un sistema BCI basato sulla P300. La componente P300 è un potenziale ERP innato nell'uomo e che si presenta come un picco positivo a circa 300 ms dopo lo stimolo. L'elicitazione avviene mediante un opportuno paradigma detto paradigma oddball. A stimoli frequenti (eventi non target o standard), si alternano stimoli rari (eventi target) in grado di elicitare la componente P300.

In questa tesi, il sistema BCI a cui si fa riferimento è quello dell'istituto IRCCS San Camillo. In particolare, l'obiettivo di questo lavoro è stato quello di effettuare delle prime prove di validazione off-line di due algoritmi single trial basati sulla stima bayesiana. Il primo metodo è indicato con B2S, Bayesian Two Step, e si basa appunto su due step. Il primo riguarda il calcolo dell'ERP medio, equivalente al profilo atteso della P300. Il secondo riguarda, invece, la determinazione delle stime single-trial vera e propria. La seconda metodologia è indicata con MTL, Multi Task Learning, e si basa sull'ipotesi che l'insieme di tutte le sweeps acquisite da un soggetto costituisca una popolazione omogenea. L'informazione congiunta di tutto il dataset viene utilizzata per stimare simultaneamente l'ERP medio e gli ERPs single-trial.

Queste metodologie sono state applicate in primis ai dati simulati ed in secundis ai dati reali.

L'aspetto a favore dell'analisi sui dati simulati è il fatto che si conosce esattamente la componente P300 vera, per cui si può analizzare la stima ottenuta valutando quanto questa si discosti dal segnale grezzo e come stimi la componente vera. I risultati ottenuti sono soddisfacenti. Il metodo B2S riesce ad individuare bene la P300, fornendo delle stime molto regolari. Anche il metodo MTL fornisce delle buone stime, con un andamento molto simile a quello vero. La latenza viene stimata sempre bene da entrambi i metodi e questo è confermato dagli ottimi valori del coefficiente di correlazione tra latenze vere e latenze stimate, soprattutto per il metodo MTL. Non è così per la stima dell'ampiezza.

Entrambi i metodi fornisco delle stime non sempre soddisfacenti, in particolare i valori stimati con la tecnica B2S sono quasi sempre più bassi rispetto a quelli veri.

Per quanto riguarda i dati reali le stime ottenute con i due metodi sono soddisfacenti e rappresentano, in generale, un buon compromesso tra l'ERP medio e i dati della sweep corrente. Visivamente, non si è rilevata nessuna particolare differenza nelle stime per i soggetti sani ed i soggetti ALS.

Ai fini del sistema BCI non è sufficiente eseguire un confronto sulla bontà delle stime. Essendo infatti il fine ultimo il miglioramento della performance del classificatore, bisogna valutare se e quanto le tecniche proposte sono utili in questo senso.

A questo scopo, sono stati valutati tre diverse tipologie di errori. Questi sono l'errore sul riconoscimento della P300 (ep3), l'errore sul non riconoscimento della P300 (enp3) e l'errore totale (etot). Questi indici sono stati calcolati con un opportuno schema di addestramento. Poiché l'elaborazione dei metodi è destinata unicamente alle sweeps target, è stata prestata particolare attenzione all'errore sul riconoscimento della P300 (ep3).

Sui dati simulati gli errori sono risultati molto bassi e confrontabili per le due tecniche. Lo stesso vale per i dati reali. In particolare, non è stato possibile stabilire quali dei due metodi fornisca le migliori performance. Infatti, alcuni soggetti hanno errori più bassi con MTL, altri con B2S, altri ancora hanno errori confrontabili.

Gli errori ep3 ottenuti con le due tecniche proposte sono stati infine confrontati con quelli ottenuti con il sistema attualmente in uso al San Camillo che utilizza la decomposizione ICA. Entrambe le tecniche, MTL e B2S, presentano errori decisamente più bassi rispetto alla decomposizione ICA, per cui si può concludere che entrambe le tecniche sono candidate per ulteriori analisi di validazione off-line nonché di implementazione on-line.

8.2 Sviluppi futuri

Il lavoro svolto in questa tesi ha costituito una fase preliminare di validazione off-line e chiaramente ha sollevato alcuni problemi ancora aperti. In questa sezione si discutono alcuni aspetti, in modo da dare importanti spunti per possibili sviluppi futuri. Tali analisi sono state affrontate nelle tre successive sezioni.

8.2.1 Miglioramento della stima single trial con il metodo B2S

La prima analisi è inerente al miglioramento della stima B2S. Infatti, come illustrato nei casi particolari del capitolo 6, sezione 6.2.1, alcune sweeps stimate presentano un andamento smooth, a tal punto che anche in presenza di una componente ERP, la sweep stimata risulta essere molto piatta. Quest'aspetto può essere migliorato fornendo un prior differente all'algoritmo. Infatti, come spiegato, la stima B2S si basa su due step: il primo riguarda il calcolo dell'ERP medio, il secondo usa quest'ultimo per ottenere le stime single-trial finali. Il prior a cui si fa riferimento riguarda esattamente il primo step. Infatti il vettore di riferimento μ della formula 3.2.8 è stato calcolato considerando nella formula 3.2.2 generale dello stimatore il vettore medio nullo. In realtà, è possibile già in questo passo sfruttare un'importante informazione a priori, ovvero fornire come vettore

di riferimento la media filtrata di tutte le sweeps del dataset.

Con questa procedura chiaramente si modifica il vettore di riferimento μ e quindi anche le singole stime. Si propongono degli esempi di stime B2S nel caso reale, confrontando come variano le stime delle medesime sweeps ottenute con prior differenti. Per comodità si indica il caso analizzato nella tesi con l'acronimo di *ref 1*, mentre il caso appena citato viene identificato con l'acronimo *ref 2*.

In figura 8.2.1 si mostrano degli esempi di stima di sweeps target relative ad un soggetto sano.

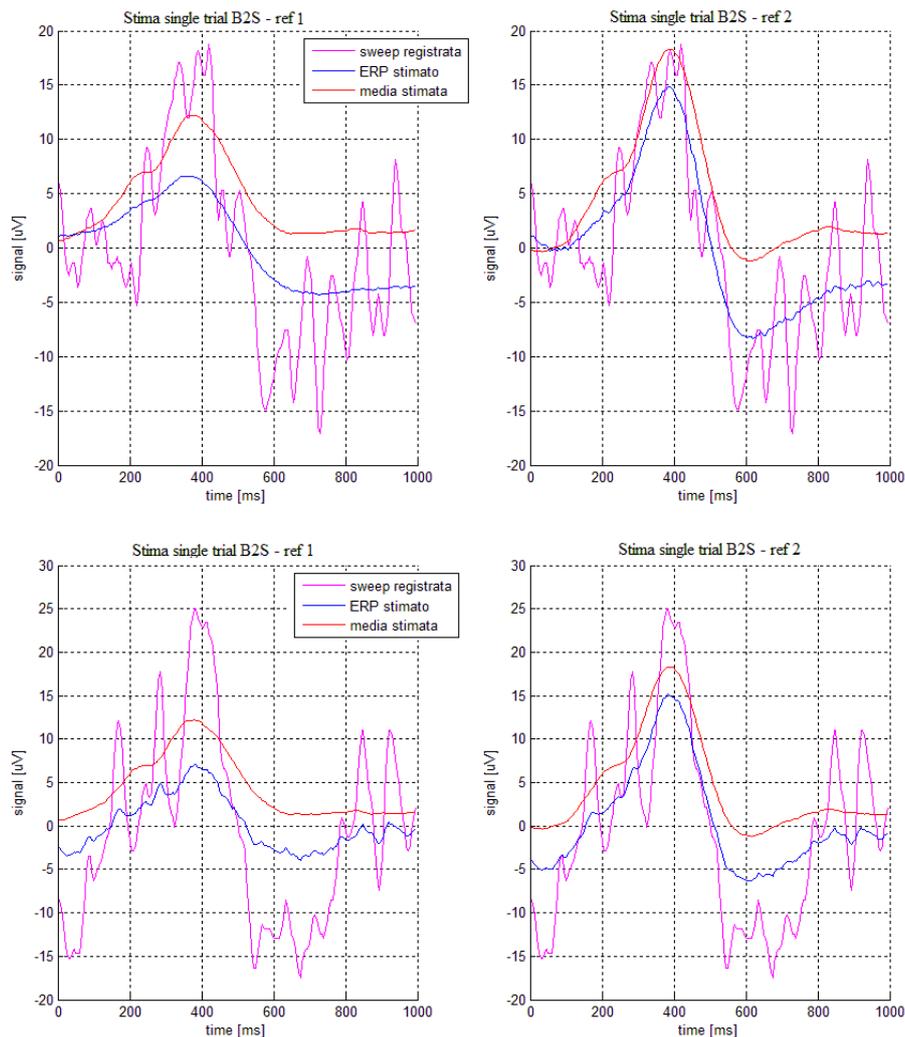


Figura 8.2.1: Confronto tra le stime con il metodo B2S con prior *ref 1* (*sx*) e *ref 2* (*dx*). Si mostrano due sweeps del soggetto sano (in magenta la sweep registrata, in blu la curva stimata, in rosso la curva media). Con il prior *ref 2* aumenta l'ampiezza della stima, rendendo più evidente rispetto al caso *ref 1*, la localizzazione della componente ERP.

Si noti come il miglioramento con il prior *ref 2* sia netto. Infatti, entrambe le stime

ottenute con prior *ref 1* mostrano l'effetto di oversmoothing, con la stima di una componente ERP di ampiezza poco più di $5 \mu V$. In figura 8.2.2, si mostrano i grafici per un soggetto ALS.

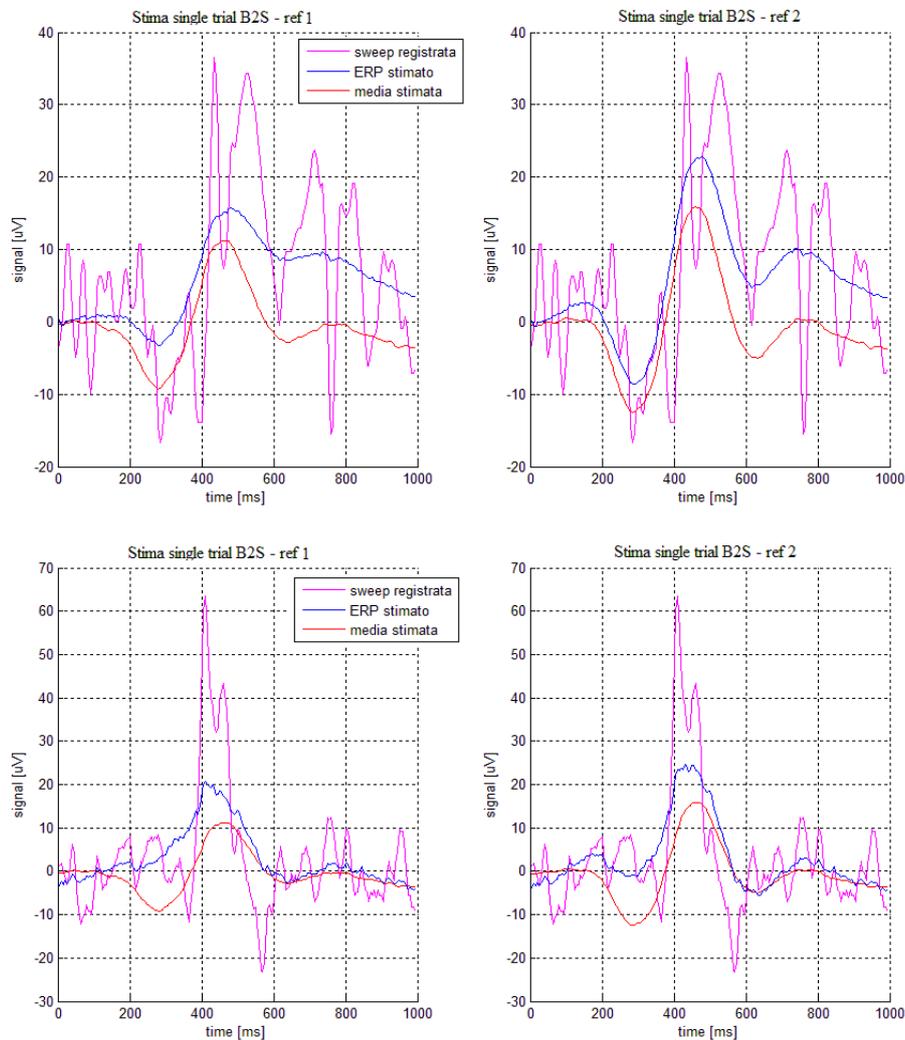


Figura 8.2.2: Confronto tra le stime ottenute con il metodo B2S con prior *ref 1*(*sx*) e *ref 2*(*dx*). Si mostrano due sweeps di un soggetto ALS (in magenta la sweep registrata, in blu la curva stimata, in rosso la curva media). Con il prior *ref 2* aumenta l'ampiezza della stima, rendendo più evidente rispetto al caso *ref 1* la localizzazione della componente ERP.

Si noti come la modifica del prior nel calcolo del vettore di riferimento μ generi nella stima un aumento dell'ampiezza della componente ERP stimata. In particolare, nel pannello superiore, l'ampiezza della sweep stimata presenta un incremento netto rispetto alla sweep del medesimo soggetto rappresentata nel pannello inferiore. Infatti, la maggior parte delle sweeps presentano un comportamento simile a quello mostrato nel pannel-

lo inferiore mentre le restanti presentano un incremento significativo dell'ampiezza della P300, come nel pannello superiore di figura 8.2.2. Valutando le stime ottenute con il prior *ref 2* i miglioramenti sono evidenti. L'ampiezza è nettamente più elevata e ripropone esattamente la componente P300. Si può concludere perciò che una valida alternativa per il miglioramento della stima B2S è sicuramente l'introduzione del nuovo prior per la stima del vettore di riferimento μ .

8.2.2 Analisi della stime da sweeps non target con i metodi B2S ed MTL

La seconda analisi condotta riguarda la valutazione delle stime sia del metodo B2S che del metodo MTL ottenute dalle sweeps non target (alla luce di quanto detto in precedenza, il metodo B2S viene applicato con prior *ref 1*). Ciò che si vuole così valutare è se queste metodologie sono in grado di fornire delle stime che ben approssimano una teorica sweep non target. Tale sweep dovrebbe avere un andamento molto regolare, con l'assenza di qualsiasi picco di ampiezza evidente nel range fra i 250 - 600 ms dopo lo stimolo. Queste prove sono state condotte sia sui dati simulati che su quelli reali, considerando solo sweeps non target. I risultati nel caso della stima B2S per i dati simulati sono riportati in figura 8.2.3.

I risultati sono davvero soddisfacenti. Si noti come tutte le sweeps non target vengono stimate per come si era ipotizzato: curve piatte, con ampiezza di pochi μV ma soprattutto senza alcun picco significativo fra i 250-600 ms dopo lo stimolo che possa ricondurre alla presenza della P300.

In figura 8.2.4 sono mostrate le stime ottenute applicando il metodo MTL alle sweeps simulate non target.

Anche in questo caso la stima è ottima. Le curve sono tutte smooth, con ampiezze piccole e senza l'occorrenza del picco relativo alla componente P300.

Gli ottimi risultati ottenuti con i dati simulati sono rassicuranti. E' da ricordare però che queste sono sweeps costruite ad hoc, con solo rumore di fondo e senza la componente ERP.

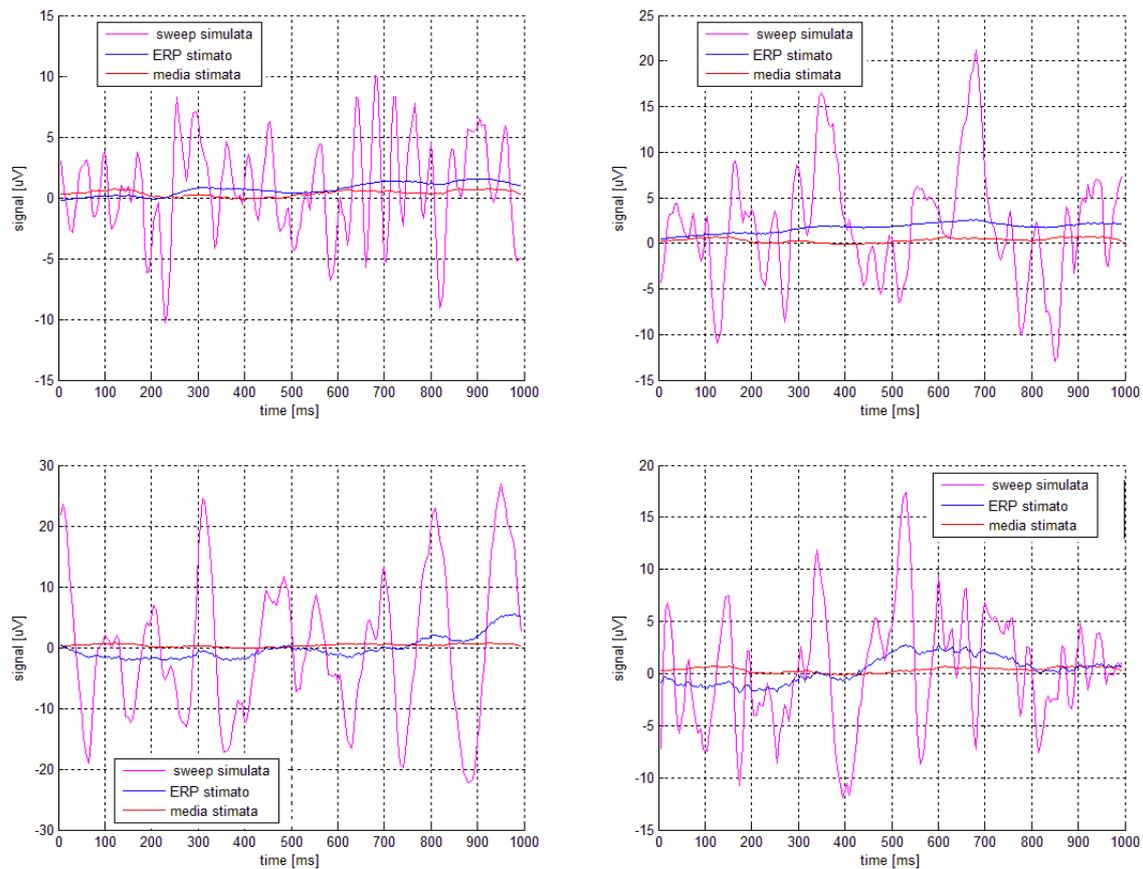


Figura 8.2.3: *Stima single trial con il metodo B2S per sweeps simulate non target (in magenta la sweep simulata, in blu la curva stimata, in rosso la curva media stimata).*

Si prosegue così con l'analisi delle stime dai dati reali. In questo caso, si analizzerà quanto i metodi riescono a limitare la stima di un'eventuale componente ERP presente nella sweep non target sulla base dell'informazione fornita.

In figura 8.2.5 si mostrano degli esempi relativi ad un soggetto sano.

Nel pannello superiore si hanno delle stime B2S ed MTL molto buone. Le curve sono piatte e non è stimata alcuna componente ERP. Si noti però che la sweep scelta non presenta dei picchi significativi. Nel pannello inferiore si mostra un caso in cui invece entrambi gli algoritmi stimano la componente P300 nella sweep corrente. La stima MTL ha un tratto marcato e un'ampiezza stimata nettamente superiore alla stima del B2S, stima che sembra ricadere nei casi limite fra sottostima dell'ampiezza ed oversmoothing. Questo aspetto non è un caso isolato. Molte sweeps relative al soggetto sano mostravano delle stime nettamente migliori per il B2S che non per MTL. A questo proposito si consideri la figura 8.2.6.

In figura 8.2.7 si mostrano degli esempi per un soggetto ALS.

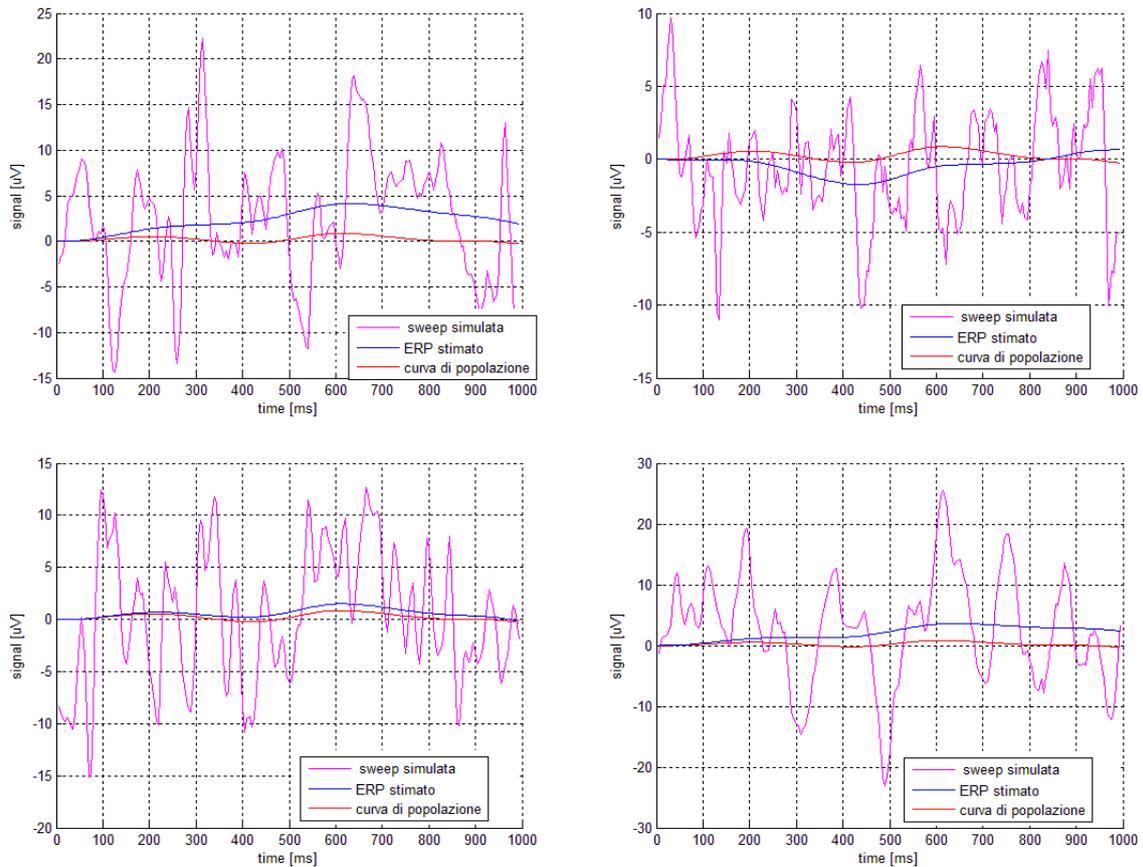


Figura 8.2.4: *Stima single trial dal metodo MTL per sweeps simulate non target (in magenta la sweep simulata, in blu la curva stimata, in rosso la curva media).*

Nel pannello superiore è mostrato un caso in cui la sweep non target stimata ha proprio l'andamento che ci si aspetta di ottenere: una curva piatta senza nessun picco significativo fra i 250-600 ms. Nel pannello inferiore, però, si mostra un caso in cui entrambe le metodologie rilevano una componente ERP nella sweeps non target. Questa condizione è quella che crea dubbi nella possibilità di applicare queste metodologie anche nel caso di sweeps non target.

La stima B2S, più regolare rispetto a quella MTL, tende a stimare meglio l'andamento delle sweeps non target. Inoltre, l'informazione a priori del B2S è sicuramente più forte rispetto al caso MTL (nel caso B2S viene costruito un vettore di riferimento ad hoc, mentre nel caso MTL l'algoritmo estrae la massima informazione sulla base del dataset che ha a disposizione) e ciò spiega gli ottimi risultati B2S ottenuti in questo caso.

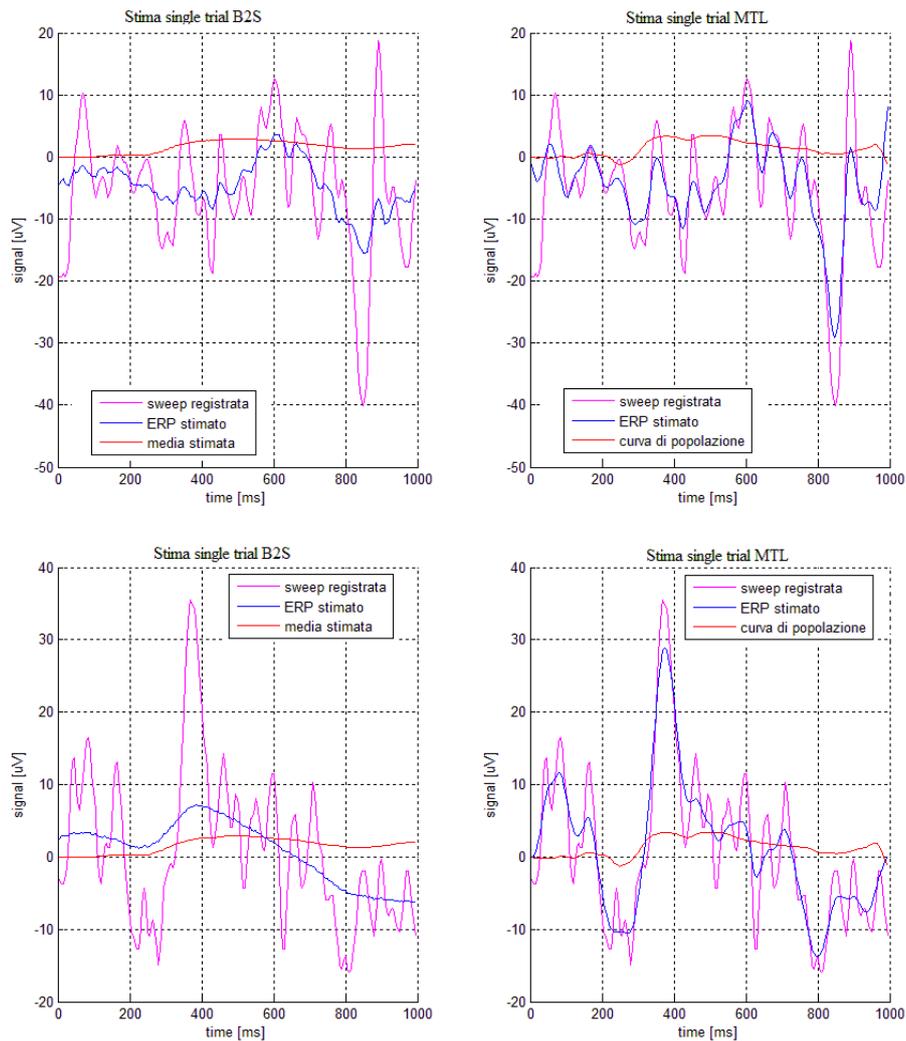


Figura 8.2.5: Esempi di stima single trial dal metodo B2S e dal metodo MTL per le medesime sweeps non target di un soggetto sano (in magenta la sweep registrata, in blu la curva stimata, in rosso la curva media). Nel pannello superiore entrambe le stime sono soddisfacenti, in quanto hanno un andamento piatto senza picchi significativi. Nel pannello inferiore è mostrata una sweep non target in cui entrambi i metodi stimano un picco significativo a circa 400 ms.

In definitiva, si può affermare che la buona stima delle sweeps non target è legata sicuramente all'andamento delle sweeps stesse. Inoltre è anche evidente, in questo caso, una variabilità intersoggettiva. Quindi, questi sono sicuramente aspetti che vanno approfonditi con altre prove per entrambi gli algoritmi.

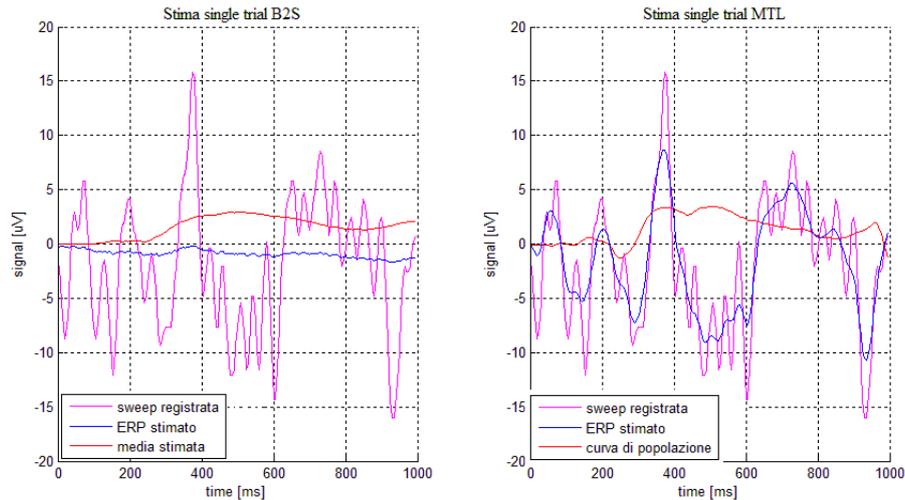


Figura 8.2.6: Esempio di stima single trial buona per il metodo B2S e non buone per il metodo MTL (in magenta la sweep registrata, in blu la curva stimata, in rosso la curva media).

8.2.3 Analisi delle performance con un nuovo schema di addestramento

L'ultima analisi che viene condotta è legata ad entrambe le analisi precedenti effettuate. Si vuole così tradurre in termini di performance quanto detto in precedenza in termini di stima. In particolare, in riferimento al soggetto sano di controllo S1 ed al paziente ALS M5, si sono valutati i tre indici di errore addestrando però il classificatore con uno schema di addestramento come indicato in figura 8.2.8.

In questo caso si applicano separatamente alle sweeps target e non target i due metodi. La differenza con quanto visto in precedenza è nell'elaborazione delle sweeps non target. La procedura di features extraction e di addestramento del classificatore, nonché la creazione del learning e testing set, rimane invariata.

Inoltre, si è voluto analizzare anche il caso in cui le sweeps target sono stimate dall'algoritmo B2S con ref 2.

Si analizzeranno così quattro condizioni

1. Applicazione del metodo MTL alle sweeps target e non target;
2. Applicazione del metodo B2S con prior ref 1 alle sweeps target e non target;
3. Applicazione del metodo B2S con prior ref 2 alle sweeps target e con prior ref 1 alle non target;
4. Applicazione ICA alle sweeps target e non target (sistema attualmente in uso al San Camillo).

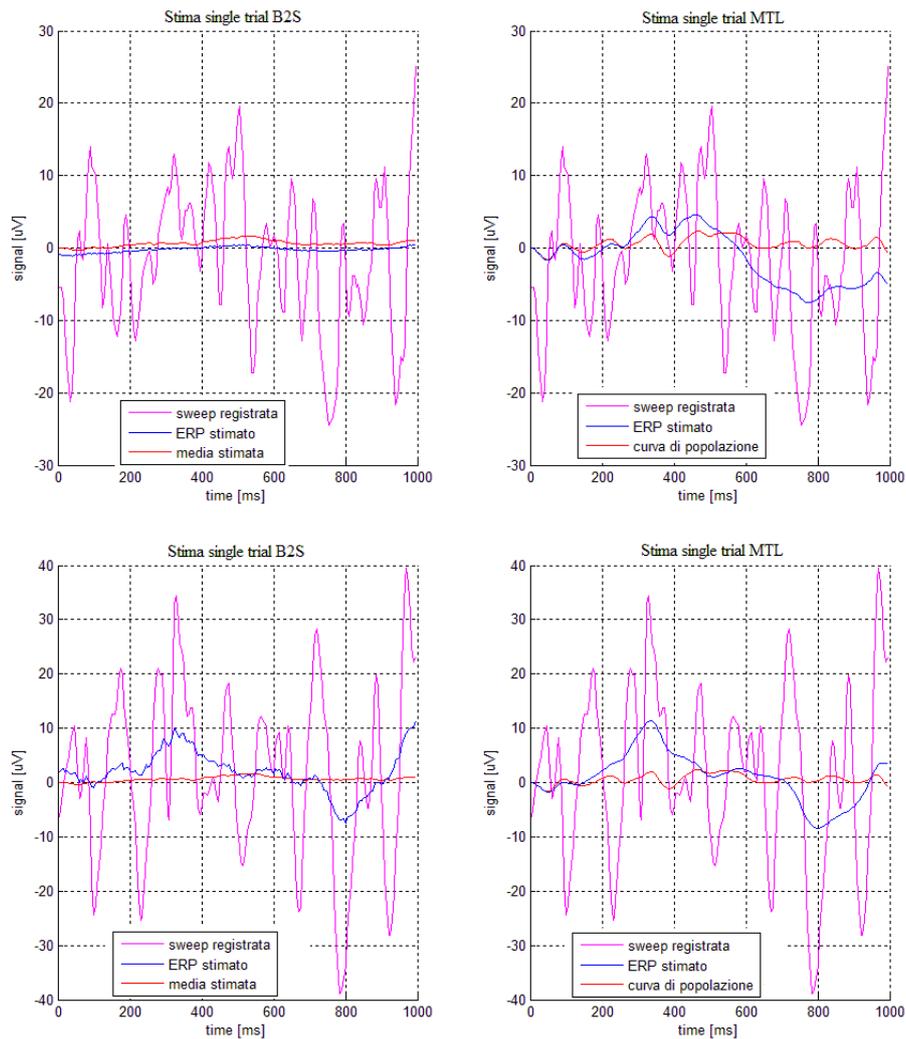


Figura 8.2.7: Esempi di stima single trial ottenute dal metodo B2S e dal metodo MTL per le medesime sweeps non target di un soggetto ALS (in magenta la sweep registrata, in blu la curva stimata, in rosso la curva media). Per entrambe le metodologie c'è una forte dipendenza dall'andamento della sweeps. Se una sweep presenta particolari oscillazioni assimilabili ad una P300 (pannello inferiore), entrambi gli algoritmi forniscono una stima che non è quella desiderata.

Sono stati addestrati 40 classificatori e sono stati calcolati gli errori medi sul testing set. Si mostrano in tabella 8.1 i risultati per il soggetto sano S1.

In generale gli errori aumentano rispetto al caso con la singola elaborazione relativa alle sweeps target ed in particolare, gli errori con il metodo MTL sono più alti rispetto al caso B2S. E' da notare invece la forte riduzione dell'errore nel caso B2S con prior ref 2 rispetto al caso con prior ref 1.

Infine, si può affermare che gli errori si mantengono più bassi o al massimo confrontabili

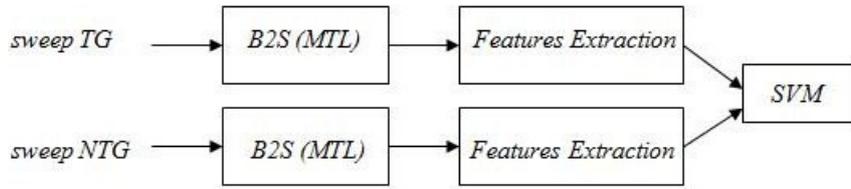


Figura 8.2.8: Schema di addestramento per il classificatore.

TESTING	B2S(ref 1)	B2S(ref 2)	MTL	ICA
ep3 (std)	0.099	0.023	0.23	0.28
enp3 (std)	0.075	0.0045	0.076	0.07
etot (std)	0.081	0.0095	0.11	0.13

Tabella 8.1: Errori medi relativi al soggetto sano S1.

con la metodologia ICA per il soggetto di controllo.

La tabella 8.2 mostra i risultati ottenuti per il soggetto ALS M5.

TESTING	B2S(ref 1)	B2S(ref 2)	MTL	ICA
ep3 (std)	0.14	0.071	0.22	0.33
enp3 (std)	0.036	0.016	0.012	0.06
etot (std)	0.055	0.025	0.049	0.11

Tabella 8.2: Errori medi relativi al soggetto ALS M5.

E' evidente la riduzione dell'errore sul riconoscimento della P300 nel caso B2S con prior ref 2 rispetto al caso B2S con prior ref 1, a conferma di quanto si era dedotto dalle stime. Sia per il soggetto sano che per quello ALS, è da notare come gli errori siano più alti rispetto a quanto si era trovato nelle tabelle finali del capitolo 7. Questo è sicuramente dovuto al fatto che adesso sia le sweeps target che non target subiscono la stessa elaborazione, che in qualche modo li rende più simili di quanto non succeda quando le sole sweeps target sono elaborate. L'aumento dell'errore sembra maggiormente a sfavore per il metodo MTL, i cui errori sono più alti rispetto al metodo B2S. Anche se si verifica questo aumento sia con il metodo B2S che MTL, tali errori rimangono comunque inferiori rispetto alla metodica ICA, confermando anche con questa modalità di applicazione un miglioramento rispetto all'attuale sistema di elaborazione.

Si può così concludere che le tecniche bayesiane, probabilmente meglio la tecnica B2S, mantengono delle ottime performance in diverse condizioni di applicazione, sia che siano solo le sweeps target ad essere elaborate, sia che l'elaborazione venga applicate ad entrambe le tipologie di sweeps. Chiaramente anche quest'ultima parte deve essere approfondita con maggiori studi e soprattutto con l'applicazione a più soggetti per valutarne le effettive performance. Ciò che si può affermare adesso è che ha senso proseguire con l'indagine di queste tecniche. Rimane solo un aspetto di fondamentale importanza da affrontare e che diventa il punto cardine di sviluppi successivi: l'implementazione on-line,

ovvero come combinare ed utilizzare l'informazione ricavata dalla stima delle sole sweeps target o anche dalle non target per la valutazione della classe di appartenenza incognita di una generica sweep. Questa è la parte che per prima dovrà essere affrontata in lavori futuri perché la scelta dell'implementazione on-line può condizionare la stessa procedura di preparazione off-line.

Per esempio, la combinazione dell'informazione della stima delle sweeps target e non target può essere ottenuta con il metodo B2S applicando on-line una sorta di pre-classificazione: eseguire un filtro matched fra la stima della sweep corrente con etichetta incognita e due template riguardanti le sweeps target e non target ricavate in fase off-line. La classe scelta sarà quella per cui risulta più elevato tale coefficiente di correlazione.

Per il metodo MTL esistono già in letteratura alcuni esempi d'implementazione on-line, come mostrato in [29], quindi mancherebbe solo la parte di combinazione dell'informazione target e non target.

Una volta implementate le tecniche anche on-line, bisognerà validarle per la fase di funzionamento on-line del sistema BCI. Questa consiste in un protocollo sperimentale, per esempio come quello spiegato nel capitolo 2, in cui soggetti sani e patologici eseguono delle prove, con vere e proprie acquisizioni on-line del segnale EEG e l'intero sistema BCI funzionante.

Appendice A

Analisi delle stime MTL con filtraggio a 15 Hz dai dati Simulati

In questa sezione si condurrà la stessa analisi della sezione 5.3, in cui però l'algoritmo MTL è applicato alle sweeps target filtrate passabasso a 15 Hz.

Il filtraggio delle sweeps passabasso con frequenza di taglio a 15 Hz comporta l'eliminazione di tutte le frequenze superiori ai 15 Hz, per cui l'effetto sarà quello di eliminazione di parte del rumore. Quanto detto è visibile nella figura A.0.1. In rosso è la sweeps grezza, mentre quella nera è la sweeps in rosso filtrata passabasso a 15 Hz.

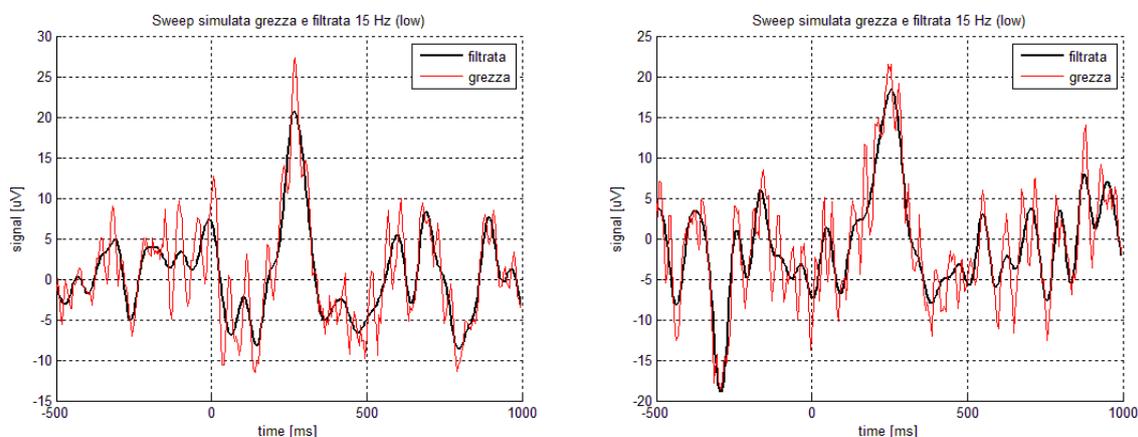


Figura A.0.1: Esempio di sweeps simulate grezze, in linea rossa, e filtrate passabasso a 15 Hz, in linea nera.

Effettivamente, la curva nera ha un andamento più regolare rispetto alla curva rossa. In modo analogo a quanto visto nella sezione 5.3, si considerano le stime dell'ERP medio e degli ERPs single-trial.

In figura A.0.2 sono riportati i risultati relativi alla stima dell'ERP medio per i dati simulati. La media vera è indicata in linea verde, la media ottenuta mediante averaging convenzionale in linea blu e la curva di popolazione stimata in linea rossa.

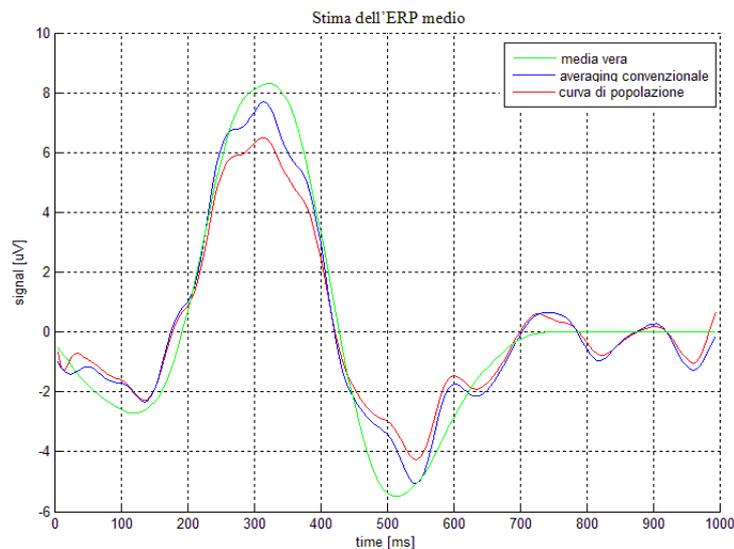


Figura A.0.2: *ERP medio dai dati simulati. In verde la media vera, in rosso la curva di popolazione ed in blu l'averaging convenzionale.*

Sia la media stimata che quella ottenuta con l'averaging convenzionale forniscono una buona stima della media vera anche se l'ampiezza del picco principale risulta sottostimata, soprattutto dal metodo MTL. In figura A.0.3 si valutano i grafici delle stime single trial ottenute dal metodo MTL. La P300 vera è indicata in linea verde, la sweep simulata in linea magenta, l'ERP stimato in linea blu e la curva di popolazione in linea rossa.

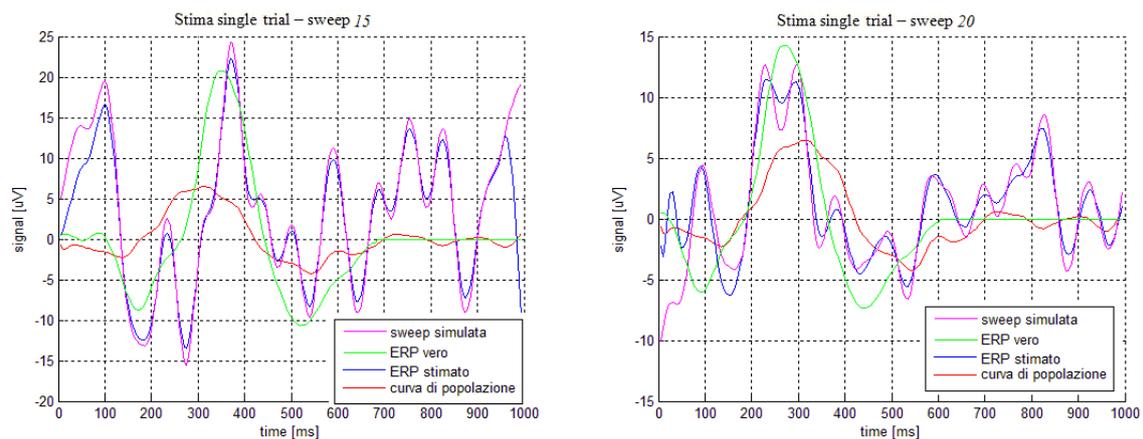


Figura A.0.3: *Stime single trial ottenute dal metodo MTL dai dati simulati.*

Si nota in questo caso un effetto non notato in precedenza in modo così evidente: la curva stimata insegue la curva simulata, mantenendone anche la forma tipica, come nel caso,

per esempio, della sweep 20 (grafico in basso a destra).

Per quanto riguarda ampiezza e latenza della P300, sono state calcolate le correlazioni tra valori veri e valori stimati. La figura A.0.4 mostra i grafici di correlazione per latenze (pannello a sinistra) e ampiezze (pannello a destra).

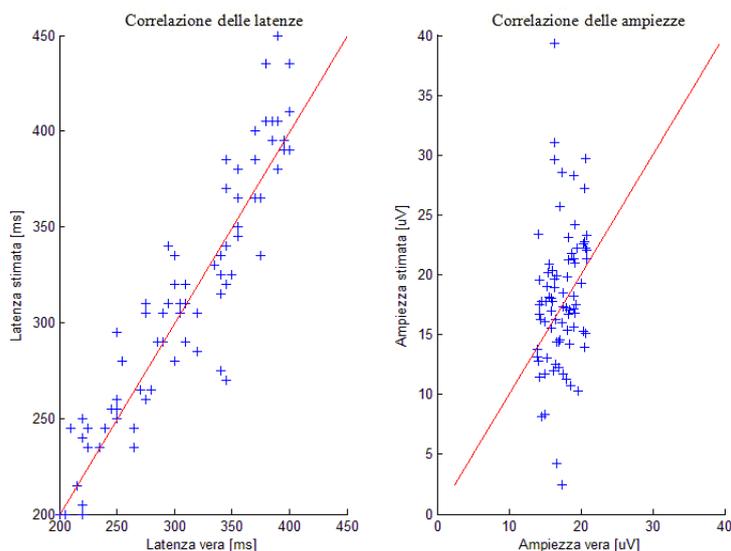


Figura A.0.4: *Correlazione delle latenze (sx) e delle ampiezze (dx)*

La correlazione della latenza è molto buona: i punti si distribuiscono lungo la retta a 45° a conferma della buona stima della latenza.

Per la correlazione dell'ampiezza invece si ripresenta lo stesso difetto riscontrato nella sezione 5.3. Un accenno di distribuzione lungo la retta a 45° si ha nella parte centrale, mentre sono evidenti ed anche numerosi i valori sottostimati e sovrastimati rispetto a quelli veri.

Il valore di correlazione ed il relativo p-value per la latenza ed ampiezza sono riportati in tabella A.1.

	correlazione	p-value
latenza	0,9277	$4,02e^{-35}$
ampiezza	0,2555	0,02

Tabella A.1: *Valori di correlazione e relativo p-value per la latenza e per l'ampiezza.*

Nel caso della latenza, il valore di correlazione è elevato con il relativo p-value molto basso a conferma della sua significatività. Per l'ampiezza, invece, la correlazione ha un valore molto basso ed un p-value elevato.

Infine si visualizzano gli istogrammi relativi alle latenze ed ampiezze in figura A.0.5 e A.0.6.

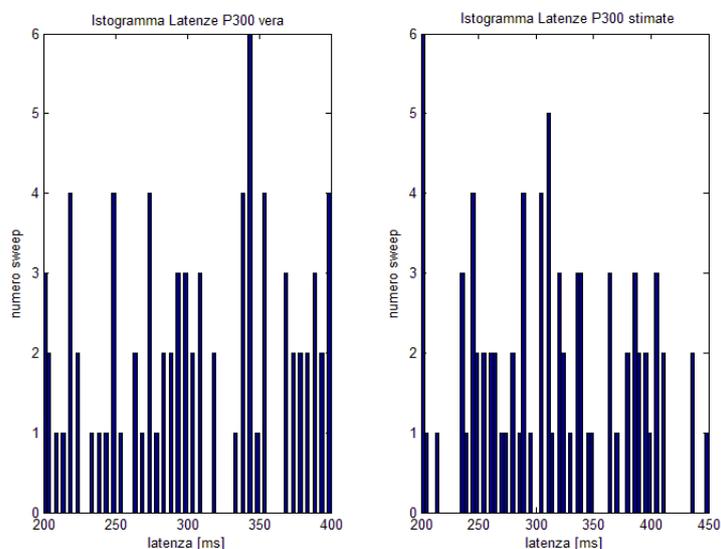


Figura A.0.5: *Istogramma delle latenze vere (sx) e delle latenze stimate (dx)*

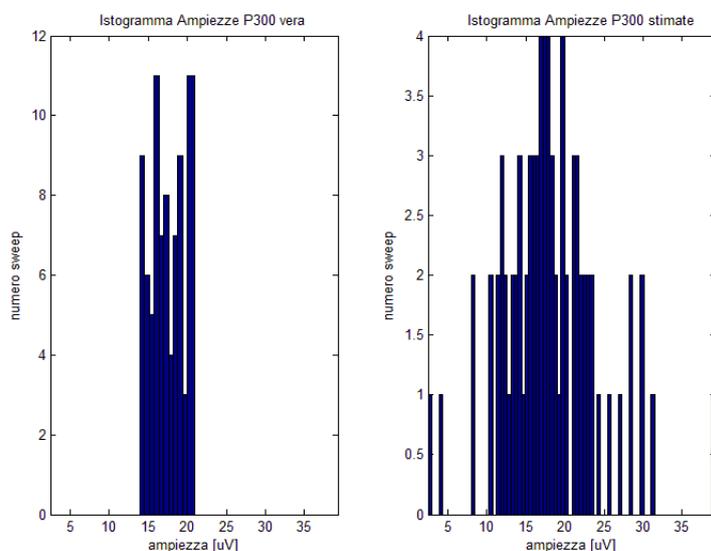


Figura A.0.6: *Istogramma delle ampiezze vere (sx) e delle ampiezze stimate (dx)*.

Il range delle latenze è pressoché identico a quello delle latenze vere (pannello di sinistra), anche se nel caso stimato sono presenti alcuni valori oltre il limite dei 400 ms; anche la distribuzione dei valori nel range dei due istogrammi è del tutto confrontabile.

Per quanto riguarda le ampiezze, il range di valori stimati è ben più grande rispetto a quello vero. C'è sicuramente una maggiore distribuzione fra i 10 e 20 μV , ma molti sono i

valori inferiori e superiori a conferma della rispettiva sottostima e sovrastima rispetto al caso vero.

Appendice B

Codice Matlab

In questa parte vengono introdotti i principali file Matlab creati ed utilizzati in questo lavoro. In particolare si inizia dal vedere il codice relativo al processo di simulazione dei dati.

Si ricorda che il soggetto scelto è M5 perché quello che ha la template (media di tutte le sweeps) più evidente. Quindi per la simulazione, è necessario per prima cosa identificare i modelli AR nella parte di pre-stimolo relativa al medesimo soggetto e memorizzare così solo quelli stabili e bianchi. La funzione che realizza ciò è la *simulaData.m*:

```
%% SIMULA DATA
clear all
close all
clc

%carico i dati del soggetto scelto
load('sbjM5')
%canale selezionato: Pz
sch = 3;
%seleziono le sweeps target e non target dal canale Pz
[sweep1,sweep2] = getSelectChanelSweep(M5.ss1,M5.ss2,M5.ai.nch,sch);
[h] = plot_MAT_avg(M5,sch);
%pre-trigger sample
pre_sweep2 = sweep2(1:M5.ai.pre,:);
%% create AR model
%range ordine modello
p_min = 2;
p_max = 14;
criterio = 'AIC';
%parametri Test Anderson
K=25;
alpha=0.05;
%AR model
```

```
nr = 1;
%Per ogni pre-stimolo si identifica il modello AR ottimo
for k = 1 : size(pre_sweep2,2)
    fprintf('File %i\n',k)
    s = pre_sweep2(:,k) - mean(pre_sweep2(:,k));
    [coeff,sigmav,bianco,stabile] = ar_opt(s,p_min,p_max,criterio,K,alpha);
    if(bianco && stabile)
        eval(['AR' num2str(nr) '.a = coeff;']);
        eval(['AR' num2str(nr) '.noisev = sigmav;'])
        eval(['save modelAR' num2str(nr) ' AR' num2str(nr)]);
        nr=nr+1;
    end
end
```

Quindi per ogni pre-stimolo si applica la funzione *ar_opt.m*, la quale identifica così il modello AR e ne valuta la bianchezza e stabilità:

```
function [coeff,sigmav,bianco,stabile] = ar_opt(s,p_min,p_max,criterio,K,alpha)
%AR_OPT 6/12/2010
%
%calcola il modello AR relativo alla traccia x.
%input:
%-'s'      :   sweep di prestimolo
%-'p_min'  :   ordine minimo
%-'p_max'  :   ordine massimo
%-'criterio' :   criterio di parsimonia
%-'K'      :   numero rispetto cui valutare la funzione di correlazione
%-'alpha'  :   livello di significativita.
%
%output
%-'coeff'  :   coefficienti del AR
%-'sigmav' :   varianza del rumore
%-'bianco' :   flag bianchezza rumore (1 si, 0 no)
%-'stabile':   flag stabilita sistema (1 si, 0 no)
%
%by C.Genna (December 2010)

if(nargin < 5)
    K=25;
    alpha = 0.05;
end
x = s - mean(s);
```

```

ct = 1;
grid = p_min : p_max;
for p = p_min : p_max
    th = ar(x,p,'yw');
    switch criterio
        case 'AIC'
            %AIC
            value(ct) = aic(th);
        case 'FPE'
            value(ct) = fpe(th);
    end
    ct = ct+1;
end
%Ordine ottimo
[val,ind] = min(value);
op = grid(ind);
%Optimum AR
Sth = ar(x,op,'yw');
coeff = Sth.a;
sigmav = Sth.noisevariance;
%verifica stabilita del sistema
stabile = isStable(Sth.a);
en = filter(Sth.a,1,x)/sqrt(Sth.noisevariance);
%applica Test di bianchezza
bianco= testAnderson(en,K,alpha);
end

```

I modelli AR così creati sono salvati in una cartella specifica. A questo punto, è possibile generare le sweeps simulate. Due sono le tipologie e quindi le funzioni che le generano. Si inizia dal processo di simulazione delle sweeps non target mediante la funzione *generaSweepNTG.m*:

```

function [sweep1] = generaSweepNTG(nNTG,nAR,const)
%GENERASWEEPTARGET 6/12/2010
%
%input:
%-'nNTG'      :   numero epoche non target da creare
%-'nAR'       :   numero modelli AR
%-'const'     :   fattore di scala per il rumore
%
%output:
%-'sweep1'    :   matrice delle sweep NTG generate.
%
%by C.Genna (December 2010)

```

```
sweep1=[];
N=400;
grid = 1: nAR;
for k = 1 : nNTG
    sAR = randsample(grid,1);
    load(['modelAR' num2str(sAR) '.mat'])
    eval(['AR = AR' num2str(sAR) ';'])
    x = const*sqrt(AR.noisev)*randn(N,1);
    tmp = filter(1,AR.a,x);
    sweep1(:,k) = tmp(101:end);
end
```

E quindi per le sweeps target:

```
function [sweep2,SNR,sweepP300,lat,amp]=generaSweepTG(nTG,nAR,t,temp,const)
%GENERASWEEPTARGET 6/12/2010
%
%input
%-'nTG'      :   numero sweep Target
%-'nAR'      :   numero modelli AR
%-'t'        :   scala temporale
%-'temp'     :   template della P300
%-'const'    :   fattore di scala per il rumore
%
%ouput
%-'sweep2'   :   matrice delle tracce simulate target
%-'SNR'      :   SNR ratio
%-'sweepP300' :   sweep con solo la P300
%-'lat'      :   latenza sweep P300 simulate
%-'amp'      :   ampiezza sweep P300 simulate
%
%by C.Genna (December 2010)

%initial setup
sweep2=zeros(length(temp),nTG);
lat = zeros(nTG,1);
amp = zeros(nTG,1);
sweepP300 = zeros(length(temp),nTG);
grid = 1:nAR;
N=400;
Agrid = -20:20;
Lgrid = -20:20;
for k=1:nTG
    sAR = randsample(grid,1);
```

```

load(['modelAR' num2str(sAR) '.mat'])
eval(['AR = AR' num2str(sAR) ';''])
x = const*sqrt(AR.noisev)*randn(N,1);
tmp = filter(1,AR.a,x);
EEGr = tmp(101:end);
%modifica ampiezza e latenza template
sa = randsample(Agrid,1);
sl = randsample(Lgrid,1);
P300 = zeros(length(temp),1);
index= find(temp ~= 0);
[val,centro] = max(temp);
for ct = centro :-1:index(1)
    P300(ct+sl) =(1+sa/100)*temp(ct);
end
for ct = centro + 1 : index(end)
    P300(ct+sl) =(1+sa/100)*temp(ct);
end
sweepP300(:,k) = P300;
sweep2(:,k)= P300 + EEGr;
%aggiorno i valori
[val,ind] = max(P300);
lat(k) = t(ind);
amp(k) = val;
PP300 = sum(abs(P300).^2);
PEEGr = sum(abs(EEGr).^2);
SNR(k) = PP300/PEEGr;
end
end

```

Per quanto concerne i dati reali, invece, ogni soggetto ha le diverse sessioni di prove effettuate salvate in un'apposita cartella. Quindi, per ogni soggetto si è creata una struttura che contenesse i dati desiderati nonché le informazioni utili. Perciò la funzione che realizza ciò è *loadDate.m*, funzione che permette così di selezionare solo i file d'interesse:

```

function [ssT1,ssT2,ai] = loadDate(dir)
%FUNCTION CARICADATI del 5/12/2010
%load data and delate the reject files
%
%input
%-'dir'      :   directory of files to select
%output:
%-'ssT1'    :   total sweep non target
%-'ssT2'    :   total sweep target
%-'ai'      :   acquisition information

```

```
%  
%by C.Genna (December 2010)  
  
%initial setup  
ssT1 = [];  
ssT2 = [];  
ai = [];  
cd(dir)  
[f3,p3] = uigetfile('.MAT','MultiSelec','on');  
Nfile = size(f3,2);  
ct = 0;  
if(~ischar(p3))  
    fprintf('No file select\n');  
    return;  
end  
cd('C:\Users\clara\Desktop\BCI\MultiTaskLearning\mfiles')  
for k = 1 : Nfile  
    if(~checkrejfiles(f3(k)))  
        load([p3,char(f3(k))])  
        ssT1 = [ssT1,ss1];  
        ssT2 = [ssT2,ss2];  
        ct = ct +1;  
    end  
end  
fprintf('Numero file %i\n',ct)  
cd(dir)  
[fil,path] = uigetfile('.MAT');  
load([path,char(fil)])  
ai.nch = oba.nch;  
ai.eogch = oba.eogch;  
ai.fs = oba.fs;  
ai.pre = - oba.xmin*oba.fs/10^3;  
ai.pnt = oba.pnt;  
ai.lab = oba.lab;  
ai.xmin = oba.xmin;  
ai.xmax = oba.xmax;  
ai.t = oba.t;  
ai.lsch = 0;  
ai.lpf = oba.lpf;  
ai.max_eogv = oba.max_eogv;  
cd('C:\Users\clara\Desktop\BCI\MultiTaskLearning\mfiles')  
end
```

La parte di applicazione dell'algorithmo B2S ed MTL non è riportata perché si tratta di funzioni che sono state implementate in precedenza e quindi inserite in un semplice

main. Invece, si presta adesso attenzione al main per la generazione del classificatore. In particolare, sia per i dati simulati, che per i dati reali, la procedura di applicazione è la medesima, per cui si valuta un singolo esempio.

Si è detto che i risultati finali sono stati ottenuti utilizzando la stima delle sweeps target ottenuta da uno dei metodi fra B2S ed MTL e lasciando le sweeps non target senza elaborazione. Successivamente si sono creati un certo numero di classificatori in modo da avere un'ampia casistica. La funzione che implementa questa parte è formalmente identica per i due metodi. Cambiano semplicemente i nomi dei file che si vanno a caricare. La funzione che si fornisce come esempio è relativa all'addestramento del classificatore sulla base delle stime MTL ottenute dai dati simulati del soggetto M5:

```
%% Applicazione SVM ai dati
clear all
close all
clc

load DatiSimulati_sbjM5_grezze
load StimaMT_datiSimulati_sbjM5_grezze
msg='c1MT_datiSim_sbjM5_grezze'
%gruppo selezionato
sng = 3;
%sweep target
ss2=squeeze(MT_estimate.ind_estimates(sng,:,:));
%numero sweep non target
ntg = size(ss2,2);
nNtg = ntg*3;
ss1 = sweep1(:,1:nNtg);

%Generazione del learning set e testing set
%numero di sweeps per learning
LnNTG = 165;
%numero di sweeps per testing
LnTG = 55;
%setup SVM
niter = 20;
cltypestr = 'SVM';
svmp.parG=0.005;
svmp.parJ=0.8;
for ct = 1 : 35
    fprintf('- SVM = %i\n',ct);
    [Lss1,Lss2,Tss1,Tss2,Lsnr,Tsnr]=generaLrTtSet(ss1,ss2,LnNTG,LnTG,MT_estimate.SNR);
    cl = create_MT_SVM(1,Lss1,Lss2,Tss1,Tss2,ai,niter,cltypestr,svmp);
    clMT.learning.ep3(ct) = cl.trainer(1);
    clMT.learning.enp3(ct) = cl.trainer(2);
end
```

```
clMT.learning.etot3(ct) = cl.trainer(3);
clMT.learning.snr(ct,:) = Lsnr;
clMT.testing.ep3(ct) = cl.testter(1);
clMT.testing.enp3(ct) = cl.testter(2);
clMT.testing.etot3(ct) = cl.testter(3);
clMT.testing.snr(ct,:) = Tsnr;
clMT.parG(ct)= cl.parG;
clMT.parJ(ct)= cl.parJ;
end
save(msg, 'clMT')
```

Per ogni classificatore, il learning e testing set viene modificato in modo random e questo viene effettuato dalla funzione *generaLrTtSet.m*:

```
function [Lss1,Lss2,Tss1,Tss2,Lsnr,Tsnr]=generaLrTtSet(ss1,ss2,LnNTG,LnTG,SNR)
%GENERALRTRSET 12/01/2011
%genera i learning set e testing set in modo random sulla base del numero
% e delle sweeps fornite
%
%input:
%- 'ss1'      :   sweeps non target
%- 'ss2'      :   sweeps target
%- 'LnNTG'    :   numero di sweeps non target di learning
%- 'LnTG'     :   numero di sweeps target di learning
%- 'SNR'      :   signal to noise
%
%output:
% - 'Lss1'    :   non target learning sweeps
% - 'Lss2'    :   target learning sweeps
% - 'Tss1'    :   non target testing sweeps
% - 'Tss2'    :   target testing sweeps
% - 'Lsnr'    :   snr sweeps learning
% - 'Tsnr'    :   snr sweeps testing

%SWEEPS NON TARGET
% number sweep non target
nNTG = size(ss1,2);
gridNTG = 1 : nNTG;
%index non target sweeps
LindexNTG = randsample(gridNTG,LnNTG);
TindexNTG = setdiff(gridNTG,LindexNTG);
Lss1 = ss1(:,LindexNTG);
Tss1 = ss1(:,TindexNTG);
```

```

%SWEEPS TARGET
% sweep target
nTG = size(ss2,2);
gridTG = 1 : nTG;
%numero sweep target
LindexTG = randsample(gridTG,LnTG);
TindexTG = setdiff(gridTG,LindexTG);
Lss2 = ss2(:,LindexTG);
Tss2 = ss2(:,TindexTG);
if(~isempty(SNR))
    Lsnr = SNR(LindexTG);
    Tsnr = SNR(TindexTG);
end
end

```

Infine si applica così il classificatore. In particolare, il SVM a cui si fa riferimento è *SVM light 5.0*, un tool di funzioni implementate nella *dll* (dynamic link library) detta *clq1a*. Le funzioni necessarie per l'applicazione del SVM sono implementate in C++. Non è necessario però scrivere function per il loro utilizzo in tale linguaggio proprio perché Matlab mette a disposizione la funzione *calllib*, la quale permette di utilizzare le funzioni C++ richiamandole nel programma Matlab.

I parametri di input della function *calllib* saranno così:

1. la *dll* a cui si fa riferimento e nel caso presente è la *clq1a*;
2. la funzione che si intende applicare;
3. i relativi parametri di input della funzione al punto 2;
4. le variabili in cui memorizzare l'output della funzione stessa.

Le variabili del punto 4 devono essere passate sottoforma di riferimenti e questo avviene attraverso la funzione *libpointer*. Per leggere in Matlab questi risultati bisogna applicare la funzione *get*, il cui primo parametro sarà il nome della variabile al punto 4 seguito dalla voce *'Value'*.

Fatte queste precisazioni, la function Matlab implementata per il SVM è *create_MTL_SVM*, i cui parametri di input sono appunto il learning set e Testing set, suddivisi entrambi in sweeps target e sweeps non target. Si ricorda infatti che la classe di appartenenza deve essere nota proprio perché si è nell'ambito della classificazione supervisionata.

All'interno di questa funzione, si eseguono i seguenti passi:

1. Calcolo delle features mediante la funzione *features_MTL* per il learning set;
2. Calcolo delle features per il testing set. Questi parametri saranno normalizzati rispetto ai valori trovati nel learning set;

3. Applicazione della function `cu_write_pat_file` in C++. Questa funzione genera un file `.pat`, formato in cui vengono salvate le features per il successivo utilizzo in altre funzioni della dll;
4. Inizializzazione dei parametri del classificatore SVM. Per questo classificatore sono due i parametri che devono essere impostati: G è un parametro fisso il cui valore è 0.005; il secondo parametro è J , il cui valore ottimo verrà cercato proprio durante l'applicazione dell'SVM fra una griglia di valori possibili. Il suo valore iniziale è 0.8; mentre la griglia di valori va da 0.6 a 2.6 con passo di 0.2.
5. Applicazione della function `cu_svm_parameters_loop2` in C++. E' la function che determina il classificatore SVM, partendo proprio dal file `.pat` del punto 3. Il classificatore così trovato viene salvato nel formato `.svm`. Inoltre vengono forniti in output i valori numerici degli errori;
6. Applicazione della function `cu_svm_test` in C++. Si applica il classificatore generato al punto 5 sul testing set, fornendo in output gli errori del classificatore sul testing set.

Nel sistema "classico" il classificatore generato era costituito da due file: `.svm`, equivalente a ciò che viene calcolato al punto 5; `.mat`, è il classificatore in formato MAT-file, formato che memorizza le principali caratteristiche del classificatore, necessarie soprattutto per l'applicazione on-line dello stesso.

Nel caso in esame, si è creato solo uno dei file del classificatore, quello `.svm`. Questo perché si sta conducendo la validazione off-line, quindi il fine è quello di ottenere unicamente le performance del classificatore. Tra l'altro, il numero di classificatori che si sono creati è molto alto e non essendo utilizzati, la creazione del secondo formato era un dispendio di memoria oltre che di tempo computazionale. Quando quindi questa tecnica vorrà essere implementata on-line, questa sarà una delle parti che dovrà essere aggiornata. Quanto detto è implementato nella seguente funzione `createSVM.m`:

```
function [c1]=createSVM(aut,Lsweep1,Lsweep2,Tsweep1,Tsweep2,ai,niter,cltypestr,svmp);  
% CREATE_BA1.M 22/10/2009  
% Create the BA1 classifier based on figure hb03 selected files;  
% if aut=1 the classifier parameters will be automatically selected;  
% input:  
% - 'aut'           :   automatic parameters selection  
% - 'Lsweep1'      :   non target learning sweeps  
% - 'Lsweep2'      :   target learning sweeps  
% - 'Tsweep1'      :   non target testing sweeps  
% - 'Tsweep2'      :   target testing sweeps  
% - 'ai'           :   informazioni d'acquisizione  
% - 'niter'        :   number of iterations to training the classifier  
% - 'cltypestr'    :   classifier type string ('SVM')  
% - 'svmp'         :   struct SVM
```

```

%
%output:
% - 'cl'           :   create classifier struct
%
% by C.Genna (Genuary 2011)

% nargin check
if (nargin ~= 9),
    error('error on input arguments');
end;
%initial setup
fs=ai.fs;
nch =ai.nch;
%number non target learning sweeps
ln1 = size(Lsweep1,2);
%number target learning sweeps
ln2 = size(Lsweep2,2);
%number non target testing sweeps
tn1 = size(Tsweep1,2);
%number target testing sweeps
tn2 = size(Tsweep2,2);
% epoch min and max time
tmin=(ai.pre+1)/1e3;
tmax=ai.pnt/1e3;
% training and testing n. of patterns
npat2 = ln1+ln2;
npat3 = tn1+tn2;
% prepare message
% fprintf(['Learning: n1=%d; n2=%d;\nTesting: n1=%d; n2=%d;\nlpf =%i\n'],...
%     ln1,ln2, tn1,tn2,lpf);
% pre-epoch points and full epoch points
pre1 = 0;
pnt1 = size(Lsweep1,1);
% get learning patterns
[Lbpat] = getfeat2(fs,nch,pre1,pnt1,Lsweep1,Lsweep2, []);
% get testing patterns
[Tbpat] = getfeat2(fs,nch,pre1,pnt1,Tsweep1,Tsweep2,Lbpat);

% load library
hffn1 = 'C:\Users\clara\Desktop\BCI\P300Protocols\clq1a\clq1a.h';
loadlibclq1a2(hffn1);
train_fname = 'ctrain1.pat';
htsvmdir = 'C:\Users\clara\Desktop\BCI\toolbox\himtools\svm';
train_ffn = [htsvmdir, '\', train_fname];

```

```
% setup SVM training parameters
DEF_PTEST = 20;
nfea = Lbpat.nfea;
wdir = [htsvmdir, '\'];
opg = libpointer('doublePtr',svmp.parG);
opj = libpointer('doublePtr',svmp.parJ);
MIN_J = 0.6;
MAX_J = 2.6;
STEP_J = 0.2;
tcnt = ((MAX_J - MIN_J) / STEP_J) + 1);
RCOLS1 = 6;      % number of columns for 'pMX' matrix (SVM training)
RCOLS2 = 4;      % number of columns for 'rMX' matrix (SVM training)
pMX = zeros(tcnt,RCOLS1)';
opmx = libpointer('doublePtr',pMX);
rMX = zeros(niter,RCOLS2)';
ormx = libpointer('doublePtr',rMX);
onxj = libpointer('int32Ptr',0);
onxe = libpointer('int32Ptr',0);
npat = size(Lbpat.v,1);
onpat = libpointer('int32Ptr',npat);
% write training .pat file
dt = datestr(now);
r = calllib('clq1a','cu_write_pat_file',train_ffn,Lbpat.v',...
    size(Lbpat.v,1),Lbpat.nfea,1,dt);
if (r ~= 0),
    error(sprintf('error create/opening file ''%s'' !\n',train_ffn));
end;
% fprintf('''%s'' file created !\n',train_ffn);
% call SVM parameters loop detection
MAX_CH_LEN = 256;
[r] = calllib('clq1a','cu_svm_parameters_loop2',aut,niter,DEF_PTEST,...
    nfea,MIN_J,MAX_J,STEP_J,MAX_CH_LEN,wdir,train_ffn,opg,opj,opmx,...
    ormx,onxj,onxe,onpat);
if (r ~= 0),
    h = warndlg(sprintf('Some error occurs during SVM training !'));
    uiwait(h);
    return;
end;
% get output parameters and results
parG = get(opg,'Value');
parJ = get(opj,'Value');
pMX = get(opmx,'Value')';
rMX = get(ormx,'Value')';
nxj = double(get(onxj,'Value'))+1;
```

```

nxe = double(get(onxe, 'Value'))+1;
npat2 = double(get(onpat, 'Value'));
trep3 = rMX(nxe,1);
trenp3 = rMX(nxe,2);
tretot = rMX(nxe,3);
% number of patterns check
if (npat2 == npat),
%   fprintf('n. of patterns:\t%d\tok !\n',npat2);
else,
    fprintf('wrong n. of patterns:\t%d\t(instead of %d)\n',npat2,npat);
    return;
end;
%
if (r ~= 0),
    error(sprintf('error SVM classifier with ''%s'' !\n',train_ffn));
end;
% set training SVM model filename
DEF_SVM_MODEL_NAME = 'model';           % default SVM model name
DEF_SVM_MODEL_EXT = '.svm';           % default SVM model name extension
% set training SVM model filename
svm_model_fname = sprintf('%s%.2d%s',DEF_SVM_MODEL_NAME,nxe,DEF_SVM_MODEL_EXT);
svm_ffn = sprintf('%s%s',wdir,svm_model_fname);
% setup SVM testing parameters
npat3 = size(Tbpat.v,1);
nfea = Tbpat.nfea;
testp = Tbpat.v';
otestp = libpointer('doublePtr',testp);
oep3 = libpointer('doublePtr',0);
oenp3 = libpointer('doublePtr',0);
oetot = libpointer('doublePtr',0);
clout = zeros(npat3,1)';
oclout = libpointer('doublePtr',clout);
perfstr='';
plen=0;
% call SVM testing function (based on 'Tbpat.v' patterns matrix)
r = calllib('clq1a','cu_svm_test',svm_model_fname,svm_ffn,'',...
    npat3,nfea,otestp,oep3,oenp3,oetot,oclout,perfstr,plen);
if (r ~= 0),
    h = warndlg(sprintf('Some error occurs during SVM testing !'));
    uiwait(h);
    return;
end;
% get results
tsep3 = get(oep3, 'Value');

```

```
tsep3 = get(oenp3,'Value');
tsetot = get(oetot,'Value');
clout = get(oclout,'Value');
% prepare classifier structure
dt = datestr(now);
dtstr = subb3(dt, '_');
directory = pwd;
clnum=1;
cl.fname = sprintf('cli_MT_svm_%.3d_%.3s.mat',clnum,dtstr);
cl.ffn = [directory, '\', cl.fname];
cl.type = cltypestr;
cl.model_fname = sprintf('cli_MT_SVM_%.3d_%.3s',clnum,dtstr,DEF_SVM_MODEL_EXT);
cl.model_ffn = [directory, '\', cl.model_fname];
cl.dt = dt;
cl.nfea = nfea;
cl.parG = parG;
cl.parJ = parJ;
cl.trainer = [trep3, trenp3, tretot];
cl.testter = [tsep3, tsep3, tsetot];
cl.pMX = pMX;
cl.rMX = rMX;
cl.nxj = nxj;
cl.nxe = nxe;
cl.trnpat = npat2;
cl.tsnpat = npat3;
cl.outc = clout;

% delete training .pat file
calllib('clq1a','cu_file_delete',train_ffn);
% % close figures
% close(h304);
% close(h305);
return; % crate_MT_SVM()
```

Bibliografia

- [1] A. Avenanti. *Neuroscienze affettive, cognitive e sociali*. slide Facoltà di Psicologia, Università di Bologna, 2008.
- [2] N. Birbaumer, A. Kübler, N. Ghanayim, T. Hinterberger, J. Perelmouter, J. Kaiser, I. Iversen, B. Kotchoubey, N. Neumann, H. Flor. *The thought translation device (TTD) for completely paralyzed patients*, IEEE, pages 8(2):190-3, 2000.
- [3] N. Birbaumer. *Breaking the silence: brain-computer interfaces (BCI) for communication and motor control*, Psychophysiology, pages 43, 517–532, 2005.
- [4] R. Caruana. *Multitask learning*, Pratt L., Thrun S. (Editors), Machine Learning, pages 28:41-75, 1997.
- [5] F. Cincotti, L. Bianchi, G. Birch, C. Guger, J. Mellinger, R. Scherer, R. Schimdt, O. Suárez, G. Schalk. *BCI-Meeting 2005-Workshop on Technology: Hardware and Software*, IEEE, vol. 14, 2006.
- [6] A. Cinetto. *Brain Computer Interface basata su P300: stato dell'arte e sviluppo di un nuovo sistema di analisi single trial*, tesi Magistrale in Bioingegneria, Padova, 2010.
- [7] B. Dal Seno, M. Matteucci, M. Mainardi. *Online-Detection of P300 and Error Potentials in a BCI Speller*, Computational Intelligence and Neuroscience, pages 5, 2010.
- [8] C. D'Avanzo, S. Schiff, P. Amodio, G. Sparacino. *A Bayesian method to estimate single-trial event-related potentials with application to the study of the P300 variability*, PubMed, 2011.
- [9] C. D'Avanzo, A. Goljahani, A. Schiavon, F. Suman, G. Pillonetto, G. De Nicolao, G. Sparacino. *A Multi-Task Learning method for the single-trial estimation of cognitive evoked potentials*, Secondo Congresso Nazionale di Bioingegneria, 2010.
- [10] E. Donchin, W. Ritter, C. McCallum. *Cognitive psychophysiology: the endogenous components of the ERP*, New York: Academic Press, pages 349–411, 1978.
- [11] Encyclopedia of Medical Devices and Instrumentation, Second Edition. *Evoked Potentials*, Webster, 2006.

- [12] L.A. Farwell, E. Donchin. *Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials*. *Electroencephalography and Clinical Neurophysiology*, pages 510–523, 1988.
- [13] M. Fatourechi, A. Bashashati, R.K. Ward, G.E. Birch. *EOG and EMG Artifacts in Brain Interface Systems: A Survey*, *Electroencephalography and Clinical Neurophysiology*, vol. 118, pages 480-494, 2007.
- [14] R. Fazel, J. Rezai. *P300 wave feature extraction: preliminary results*, IEEE, 2005.
- [15] Ganong. *Fisiologia Medica*, Piccin, 2003.
- [16] F. Grandori, A. Martini. *Potenziali Evocati Uditivi*, Piccin, pages 378-390, 1995.
- [17] P. Heinrich, M. Bach. *Signal and noise P300 recordings to visual stimuli*, *Documenta Ophthalmologica*, vol. 117, pages 73-83, 2007.
- [18] H. Kececi, Y. Degirmenci, S. Atakay. *Habituation and Dishabituation of P300*, *Cognitive & Behavioral Neurology*, vol. 19, pages 130-134, 2006.
- [19] T. Joachims. *Making Large-Scale SVM Learning Practical*, UNIDO, University of Dortmund Computer Science Department, 1998.
- [20] K. Böcker, J. van Avermaete, M. C. van den Berg-Lenssen. *The international 10–20 system revisited: Cartesian and spherical co-ordinates*, PubMed, Spring, vol 6, pages 231-235, 1994.
- [21] A. Kübler, V. Mushahwar, L. Hochberg, J. Donoghue. *BCI Meeting 2005–workshop on clinical issues and applications*, IEEE, pages 131-4, 2006.
- [22] A. Kübler, N. Birbaumer. *Brain–computer interfaces and communication in paralysis: Extinction of goal directed thinking in completely paralysed patients?*, *Electroencephalography and Clinical Neurophysiology*, vol. 119, pages 2658–2666, 2008.
- [23] A. McFarland, R. Muller, S. Dean Krisiensi. *BCI Meeting 2005- Workshop on BCI Signal Processing: Feature Extraction and Translation*, IEEE, vol. 14, 2006.
- [24] E. Niedermeyer, F. Da Silva. *Electroencephalography, basic principles, clinical applications and related fields*, Lippincott Williams &Wilkins, 2004.
- [25] F. Nijboer, EW. Sellers, J. Mellinger, MA. Jordan, T. Matuz, A. Furdea, S. Halder, U. Mochty, DJ. Krusienski, TM. Vaughan, J. Wolpaw, N. Birbaumer, A. Kübler. *A P300-based brain–computer interface for people with amyotrophic lateral sclerosis*, *Electroencephalography and Clinical Neurophysiology*, pages 1909–1916, 2008.
- [26] F. Nijboer, N. Birbaumer, A. Kübler. *The influence of psychological state and motivation on brain–computer interface performance in patients with amyotrophic lateral sclerosis – a longitudinal study*, *Frontiers in Neuroscience*, pages 1-13, 2010.

-
- [27] G. Pfurtscheller, C. Neuper. *Motor Imagery and Direct Brain-Computer Communication*, IEEE, vol. 89, 2002.
- [28] F. Piccione, F. Giorgi, P. Tonin, K. Priftis, S. Giove, S. Silvoni, G. Palmas, F. Beverina. *P300-based brain-computer interface: reliability and performance in healthy and paralysed participants*, *Electroencephalography and Clinical Neurophysiology*, pages 1-7, 2006.
- [29] G. Pillonetto, F. Dinuzzo, G. De Nicolao. *Bayesian online multi-tasklearning*, IEEE, pages 193-200, 2010.
- [30] G. Pillonetto, G. De Nicolao, M. Chierici, C. Cobelli. *Fast Algorithms for Non-parametric Population Modeling of Large Data Sets*, *Automatica*, pages 173-179, 2008.
- [31] J. Polich. *Updating P300: An integrative theory of P3a and P3b*, *Electroencephalography and Clinical Neurophysiology*, pages 2128-2148, 2007.
- [32] J. Polich, A. Joel, L. Bauer, S. Kuperman, S. Morzorati, S. Connor, B. Porjesz, J. Rohrbaugh, H. Begleiter. *P300 Topography of Amplitude/Latency Correlations*, *Brain Topography*, vol 9, pages 275-282, 1997.
- [33] D. Radven, J. Polich. *Habituation of P300 from visual stimuli*, *International Journal of Psychophysiology*, vol 30, pages 359-365, 1998.
- [34] EW. Sellers, E. Donchin. *A P300-based brain-computer interface: initial tests by ALS patients*, *Electroencephalography and Clinical Neurophysiology*, pages 117:538-48, 2006.
- [35] G. Schalk, J. McFarland, T. Hinterberger, N. Birbaumer, R. Wolpaw. *BCI2000: A General-Purpose Brain-Computer Interface (BCI) System*, IEEE, vol. 51, 2004.
- [36] M. Soskins, J. Rosenfeld, T. Niendam. *Peak-to-peak measurement of P300 recorded at 0.3 Hz high pass filter settings in intraindividual diagnosis: complex vs simple paradigms*, *International Journal of Psychophysiology*, vol 40, pages 173-180, 2001.
- [37] G. Sparacino, S. Milani, E. Arslan, C. Cobelli. *A Bayesian approach to estimate evoked potentials*, *Computer Methods and Programs in Biomedicine*, pages 68: 223-248, 2002.
- [38] G. Sparacino, C. D'Avanzo, E. Pasqualotto, E. Veronese, S. Schiff, P. Amodio. *Event Related Potentials Measurement: A Bayesian Approach to Perform Improved Averaging and Single-Trial Estimation*, *Workshop Proceedings of SIMPAR*, 2008.
- [39] S. Silvoni, C. Volpato, M. Cavinato, M. Marchetti, K. Priftis, A. Merico, P. Tonin, K. Koutsikos, F. Beverina, F. Piccione. *P300-based brain-computer interface communication: evaluation and follow-up in amyotrophic lateral sclerosis*, *Frontiers in Neuroprosthetics*, pages 1-12, 2009.

- [40] A. Schiavon. *Un metodo di Multi-Task Learning per la single-trial di potenziali evocati cognitivi*, Tesi Magistrale in Bioingegneria, Università degli studi di Padova, 2010.
- [41] G. Toffolo. *Slide del corso di elaborazione di segnali biologici*, Università degli studi di Padova, 2009.
- [42] I. Volosyak, D. Valbuena, T. Malechka, J. Peuscher, A. Graser. *Brain-computer interface using water-based electrodes*, Journal of Neural Engineering, vol 7, 2010.
- [43] JR. Wolpaw, N. Birbaumer, DJ. McFarland, G. Pfurtscheller, TM. Vaughan. *Brain-computer interfaces for communication and control*, Electroencephalography and Clinical Neurophysiology, pages 113:767–91, 2002.
- [44] JR. Wolpaw, G. Loeb, B. Allison, E. Donchin, O. do Nascimento, W. Heerderks, F. Nijober, W. Shain, J. Turner. *BCI Meeting 2005- Workshop on Signals and Recording Methods*, IEEE, vol. 14, 2006.

SITI INTERNET CONSULTATI

- [s1] <http://www.cognitiveneuroscience.it/eeg.htm>
- [s2] <http://www.la-sclerosimultipla.net>
- [s3] <http://www.sensibilab.campuspoint.polimi.it>
- [s4] <http://www.etsu.edu>
- [s5] <http://www.atlasofms.org/>
- [s6] <http://www.electropsychology.com>
- [s7] <http://spdevices.com>
- [s8] <http://www.cognitiveneuroscience.it>
- [s9] <http://cepl.nyspi.org>
- [s10] <http://www.fondazione sancamillo.org>
- [s11] <http://www.neuroscan.com>
- [s12] <http://www.emsmedical.net>
- [s13] <http://www.aitn.it/>
- [s14] <http://www.seiemg.it/EEG.htm>

Ringraziamenti

Un ringraziamento speciale va ai miei genitori, che in tutti questi anni mi hanno sempre sostenuta, nonostante tutte le difficoltà.

Ringrazio il professore Giovanni Sparacino, Costanza D'Avanzo e Anahita Goljahani per avermi seguita in questi mesi.

Ringrazio tutto lo staff dell'istituto IRCCS San Camillo, Marianna, Chiara, Andrea, Giulia per la loro allegria e per l'entusiasmo che mettono nel proprio lavoro, ma soprattutto l'ingegnere Stefano Silvoni ed il dottore Piccione, per i loro preziosi consigli sia professionali che umani.

Infine, ringrazio tutti gli amici che in questi mesi mi sono stati vicino.

Ad maiora semper.

Clara Genna