

UNIVERSITÀ DEGLI STUDI DI PADOVA

Dipartimento di Ingegneria dell'Informazione
LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

**Implementazione e validazione di una BMI
in tempo reale basata sull'entropia per
l'Intentional Non-Control**

Relatore

PROF. LUCA TONIN

Candidato

DIEGO SAPORITO

Co-Relatore

DR. STEFANO TORTORA

DR.SSA GLORIA BERALDO

17 aprile 2023

Anno Accademico 2022-2023

Abstract

Una Brain-Computer Interface (BCI) è una tecnologia che permette di usare i segnali elettroencefalografici (EEG) del cervello come mezzo di comunicazione e controllo. Nelle ultime decadi si è assistito a un incremento di complessità delle applicazioni basate su BCI, utilizzate spesso anche in ambito sanitario nel campo dell'assistenza e della riabilitazione.

Uno dei problemi più noti riguardanti lo sviluppo di applicazioni BCI è quello dell'Intentional Non-Control (INC) e cioè quello stato in cui l'utente che utilizza la BCI non ha intenzione di impartire alcun comando.

Lo scopo di questo lavoro è quello di progettare e realizzare un applicativo in grado di riconoscere lo stato di INC, sfruttando la motor imagery (MI). Per raggiungere questo obiettivo si è deciso di sfruttare l'entropia dei segnali EEG in quanto recenti studi hanno mostrato come ad attività di movimento sia associato un alto valore di entropia mentre allo stato di INC sia associato un basso valore di entropia. In particolare in questo progetto si è creato un sistema in grado di classificare tre diverse attività eseguite dall'utente: immaginazione del movimento delle mani, immaginazione del movimento dei piedi e INC.

È stato quindi sviluppato un package ROS-Neuro che, sfruttando l'entropia, sia in grado di classificare l'azione dell'utente in tempo reale.

In questa tesi saranno descritte le BCI in generale, i materiali e i metodi uti-

lizzati, l'approccio e le motivazioni dietro le scelte prese e il modello. Quindi si presenterà la configurazione sperimentale, i risultati ottenuti ed infine possibili sviluppi futuri.

Indice

Abstract	i
Indice delle Figure	vii
Indice delle Tabelle	ix
Lista degli Acronimi	xi
1 Introduzione	1
1.1 Cos'è una BCI	2
1.2 Tipologie di BCI	3
1.3 Storia e stato dell'arte	5
1.4 Problema dell'Intentional Non-Control	7
1.5 Obiettivi e struttura	8
2 Materiali e Metodi	11
2.1 Ambiente di sviluppo	11
2.1.1 Robot Operating System	11
Filesystem Level	12
Computation Graph Level	12
Community Level	13
2.1.2 ROS-Neuro	14
2.2 Scelta dell'entropia	15
2.3 Modello proposto	17

2.4	Interfaccia di training offline	19
2.5	Test e ottimizzazione	21
2.6	Struttura dell'esperimento	23
2.6.1	Dispositivo di acquisizione EEG	24
3	Risultati	27
3.1	Output dei test	29
3.2	Risultati offline	33
3.3	Risultati online	37
4	Conclusioni	43
4.1	Discussione dei risultati	43
4.2	Conclusioni e sviluppi futuri	44
	Bibliografia	47

Indice delle Figure

1.1	Struttura di una BCI	3
1.2	Disposizione degli elettrodi secondo il sistema 20-10	4
2.1	Rappresentazione schematica dell'architettura di ROS-Neuro .	15
2.2	Interfaccia mostrata all'utente nell'esperimento	17
2.3	Rappresentazione schematica dell'approccio usato per l'elaborazione e la classificazione dei segnali. Le ramificazioni indicano le diverse bande utilizzabili.	18
2.4	Applicativo utilizzato per il training del classificatore	20
2.5	Schermata di selezione delle features. In alto i Fisher score calcolati sui singoli file caricati; in basso a sinistra il Fisher score medio; in basso al centro le features selezionate.	21
2.6	(A): fase di fixation in cui l'utente viene avvertito che sta per arrivare un comando; (B): fase di riempimento della barra in cui l'utente esegue il comando indicato	24
2.7	Dispositivo di acquisizione EEG gUSBamp	25

3.1	Punteggi delle features dopo aver eseguito le acquisizioni di calibrazione. In ordine da sinistra a destra i punteggi della features (in verticale) per banda: 16-30 Hz, 2-45 Hz, 8-13 Hz, 8-35 Hz. In basso al centro è possibile vedere quali sono le features selezionate.	28
3.2	Confronto tra entropia calcolata in modo standard e tramite downsampling a 32 bins	29
3.3	Confronto tra entropia calcolata in modo standard e tramite downsampling a 64 bins	30
3.4	Confronto tra entropia calcolata in modo standard e tramite downsampling a 128 bins	30
3.5	Confronto tra entropia calcolata in modo standard e tramite downsampling 192 bins	31
3.6	Confronto tra entropia calcolata in modo standard e tramite downsampling a 256 bins	31
3.7	Confronto tra entropia calcolata in modo standard e tramite downsampling a 512 bins	32
3.8	Confronto tra i tempi di esecuzione tra i vari metodi usati per il calcolo dell'entropia	32
3.9	Confronto tra gli istogrammi con l'uso del test di Kolmogorov-Smirnov. (A: 128 bins, B: 192 bins, C: 256 bins)	33
3.10	A: Valore medio e deviazione standard dell'entropia calcolata durante l'INC offline; B: Valore medio e deviazione standard dell'entropia calcolata durante la motor imagery delle mani offline; C: Valore medio e deviazione standard dell'entropia calcolata durante la motor imagery dei piedi offline	34

3.11	Oscillazione dell'entropia media durante un trial per ogni canale durante l'acquisizione offline.	35
3.12	A: Valore medio e deviazione standard del CSP calcolato durante l'INC offline; B: Valore medio e deviazione standard del CSP calcolato durante la motor imagery delle mani offline; C: Valore medio e deviazione standard del CSP calcolato durante la motor imagery dei piedi offline.	35
3.13	Oscillazione dei valori medi del CSP calcolata durante i trials offline.	37
3.14	A: Valore medio e deviazione standard dell'entropia calcolata durante l'INC online; B: Valore medio e deviazione standard dell'entropia calcolata durante la motor imagery delle mani online; C: Valore medio e deviazione standard dell'entropia calcolata durante la motor imagery dei piedi online.	38
3.15	A: Valore medio e deviazione standard del CSP calcolato durante l'INC online; B: Valore medio e deviazione standard del CSP calcolato durante la motor imagery delle mani online; C: Valore medio e deviazione standard del CSP calcolato durante la motor imagery dei piedi online.	40
3.16	Oscillazione dell'entropia media durante un trial per ogni canale durante l'acquisizione in tempo reale.	40
3.17	Oscillazione dei valori medi del CSP calcolata durante i trials online.	41

Indice delle Tabelle

1.1	Tipologie di onde	4
3.1	Valore medio e deviazione standard dell'entropia per le acquisizioni offline.	36
3.2	Valore medio e deviazione standard dell'entropia per le acquisizioni online.	39

Lista degli Acronimi

BCI Brain Computer Interface.

CAR Common Average Reference.

CSP Common Spatial Pattern.

EEG Elettroencefalografici.

ERP Event-Related Potential.

IA Intelligenza Artificiale.

INC Intentional Non-Control.

LDA Linear Discriminant Analysis.

MI Motor Imagery.

ML Machine Learning.

ROS Robot Operating System.

SSEP Steady-State Evoked Potential.

1 | Introduzione

Il progresso scientifico e tecnologico sta permettendo sempre più passi in avanti in numerosi campi dando la possibilità di migliorare processi preesistenti o anche di raggiungere obiettivi finora considerati irrealizzabili. Tutto ciò ha permesso un grande sviluppo anche nell'ambito medico e riabilitativo con l'impiego di strumenti e macchinari a supporto del personale sanitario [1]. In particolare per ciò che riguarda la riabilitazione neurologica si è cercato di dare un ruolo più attivo al paziente per favorire la neuroplasticità [2], cioè la capacità del nostro sistema nervoso di rimodellare se stesso grazie a stimoli esterni [3].

L'obiettivo della comunità scientifica è quello di migliorare la qualità di vita delle persone affetta da disabilità motoria, sensoriale o cognitiva. Sono presenti molti studi in letteratura che si concentrano sul trattamento e la riabilitazione di pazienti affetti da malattie degenerative come per esempio il morbo di Parkinson [4][5] o la sclerosi multipla [6] oppure da infortuni di tipo traumatico come possono essere ictus [7] e amputazioni [8].

Con l'avanzare delle ricerche e degli studi nel campo dell'*intelligenza artificiale* (IA) e della *robotica* è nata una nuova branca chiamata **neurorobotica** [9].

La neurorobotica fa parte del campo delle neuroscienze e dell'ingegneria e si basa sulla combinazione delle conoscenze sul cervello umano e l'utilizzo di

tecniche di IA [10]. Si definiscono neurorobot tutti quei dispositivi in grado di comunicare, interagire e imitare il sistema nervoso umano o animale. La neurorobotica si può considerare interdisciplinare in quanto ha applicazioni di diverso tipo, da quella puramente industriale (come per esempio braccia robotiche) a quella ludica (per esempio nel mondo dei videogiochi) passando ovviamente per il settore medico dove viene sfruttata per l'appunto in ambito riabilitativo e non solo.

In merito a quanto detto uno strumento fondamentale per la neurorobotica sono le **Brain Computer Interface** (BCI).

1.1 Cos'è una BCI

Una BCI è uno strumento in grado di misurare e processare l'attività cerebrale, e identificare le intenzioni del soggetto in modo da convertirle in segnali di controllo per un dispositivo esterno come per esempio un computer, un robot o una protesi [11]. Il vantaggio di questa tecnologia sta nella possibilità per gli utenti di inviare comandi e controllare dispositivi senza l'utilizzo di alcuna attività muscolare.

Una BCI opera con una struttura di tipo *closed-loop* composta da cinque passaggi: acquisizione dei segnali, pre-processing dei segnali, estrazione delle features, classificazione e interfaccia di controllo (modulo di feedback) [12] (Figura 1.1). Il primo blocco permette l'acquisizione dei segnali provenienti dal cervello. Questa può avvenire sia tramite metodi invasivi che non invasivi. Nel secondo vengono processati in modo da poter rimuovere eventuali artefatti e migliorare il segnale da cui vengono poi estratte le features necessarie per la classificazione. Il ciclo viene invece concluso dal modulo di feedback che permette all'utente di essere a conoscenza della classificazione effettuata.

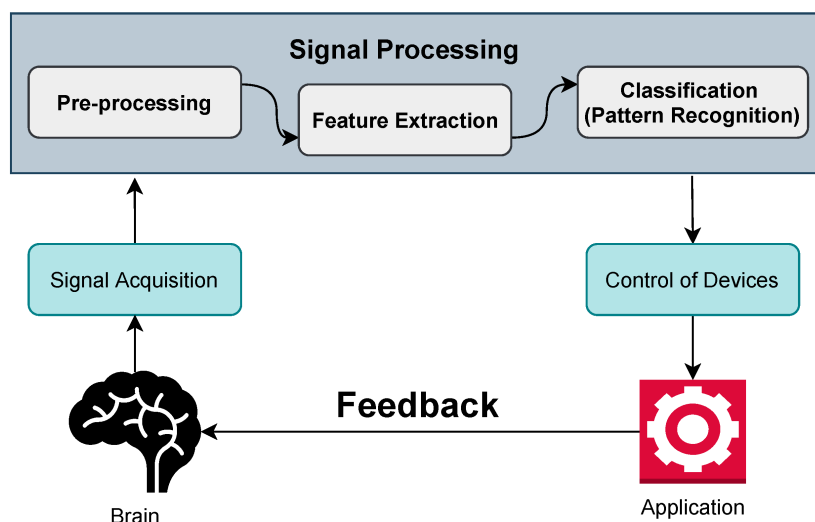


Figura 1.1: Struttura di una BCI

1.2 Tipologie di BCI

L'acquisizione dei segnali può essere effettuata in due modi: invasivo o non invasivo [13]. La differenza consiste da come e dove vengono piazzati gli elettrodi che misurano l'attività cerebrale e cioè se si trovano al di sopra della pelle (non invasiva) o meno (invasiva). Le BCI invasive utilizzano delle griglie di elettrodi impiantati direttamente nella corteccia mentre le BCI non invasive consistono solitamente in cuffie con elettrodi tramite cui è possibile misurare l'attività elettrica della corteccia, nota anche come EEG. Il vantaggio principale di quest'ultime è che non richiedono alcun tipo di intervento chirurgico per essere installate ma allo stesso tempo il segnale registrato viene deformato dalle ossa e subisce una significativa attenuazione.

Tradizionalmente, nelle BCI non invasive, gli elettrodi sono piazzati sul cuoio

capelluto sulla base del Sistema Internazionale 10-20 che è stato standardizzato dalla American Electroencephalographic Society [14].

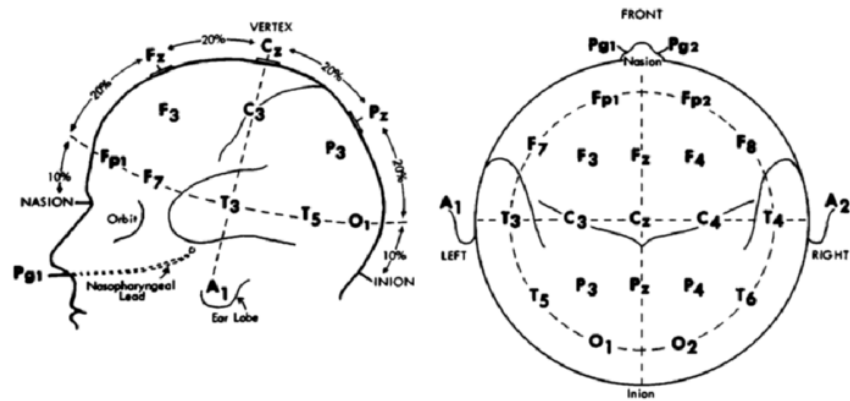


Figura 1.2: Disposizione degli elettrodi secondo il sistema 20-10

L'attività EEG registrata dal cuoio capelluto mostra delle oscillazioni, molte delle quali hanno dei caratteristici intervalli di frequenza, distribuzione spaziale a seconda del tipo di attività mentale effettuata [15] (Tabella 1.1).

Tipo	Frequenza (Hz)	Stato comportamentale/psicologico
Delta	0-4	Riposo profondo, sonno
Teta	4-8	Rilassato
Alfa	8-13	Calma
Beta	13-30	Allerta, ansia, attacco di panico, concentrazione
Gamma	30-100	Combinazione di due sensi

Tabella 1.1: Tipologie di onde

Un altro modo per distinguere le BCI è la tipologia di paradigma [16][17]. Di seguito saranno descritti alcuni di questi:

- Motor Imagery (MI): l'acquisizione viene effettuata chiedendo all'utente di immaginare un movimento piuttosto che eseguirlo realmente (per esempio movimento di entrambe le mani o entrambi i piedi). Vari studi hanno confermato che l'immaginazione attiva le aree del cervello che sono responsabili del movimento effettivo [18];
- Event-Related Potential (ERP): si basa sull'alterazione dell'attività cerebrale tramite stimoli esterni che possono essere di tipo visivo, sonoro o sensoriale. Uno dei più popolari di questa categoria è il P300 che utilizza uno stimolo di tipo visivo (Visual-Related Potential);
- Steady-State Evoked Potentials (SSEP): viene utilizzato uno stimolo visivo lampeggiante a frequenza costante che risulta in un pattern EEG coordinato con la frequenza dello stimolo. Solitamente vengono presentati al soggetto più stimoli contemporaneamente ognuno a una sua frequenza di lampeggiamento e le intenzioni dell'utente sono determinate facendo corrispondere il pattern dell'attività EEG con il comando associato a quella particolare frequenza.

È possibile inoltre combinare due o più di queste tecniche per ottenere un paradigma ibrido [19], per esempio utilizzando P300 e SSEP [20].

1.3 Storia e stato dell'arte

Se prendiamo in considerazione le onde EEG i primi studi e ricerche riconducibili alle BCI possono essere datati agli anni '20 del secolo scorso, infatti la prima attività elettrica del cervello umano è stata registrata nel 1924 e successivamente il professore universitario Hans Berger pubblicò un primo lavoro sull'encefalogramma umano nel 1929 [21]. Da quel momento in avanti

le ricerche non si sono mai fermate e grazie all'avvento dei computer negli anni '60 questi hanno reso possibile un ulteriore passo in avanti permettendo una conversione dei segnali analogici registrati in segnali digitali [22], requisito fondamentale per una BCI e la prima a comprendere questa necessità è stata la ricercatrice Telma Estrin di UCLA.

Negli anni '70 il professore James Vidal, sempre di UCLA, pubblicò quello che è ritenuto il lavoro più famoso in ambito BCI suggerendo una diretta comunicazione tra computer e cervello e sottolineando tutti gli elementi necessari per realizzare una BCI funzionante [23].

Nonostante la visione di Vidal, le limitazioni tecnologiche dell'epoca rendevano impossibile realizzare uno strumento che permettesse effettivamente di controllare una macchina tramite il cervello e nei successivi anni sono continuati gli studi e le pubblicazioni riguardo la natura di potenziale generato nel cervello sulla base di stimoli visivi o uditivi.

Grazie ai notevoli passi in avanti fatti nel corso degli anni con l'arrivo del nuovo millennio i lavori e gli studi in questo ambito si sono moltiplicati, ma la crescita delle pubblicazioni riguardanti utenti con disabilità è stata molto meno rapida.

Nel 2010 la quantità di informazione trasferibile a una BCI per unità di tempo è cresciuta incredibilmente e con tale velocità di comunicazione le BCI hanno potuto rappresentare una valida tecnologia assistiva [24].

Al giorno d'oggi le tecniche e gli approcci si sono diversificati e sono disponibili più modi per poter elaborare i dati. Per il pre-processing, quindi per la rimozione del rumore e il miglioramento del segnale, sono impiegati filtri spaziali come per esempio il Common Average Reference (CAR) [25] o il filtro Laplaciano [26], e filtri temporali come quello di Butterworth [27] o di Chebyshev [28].

Per l'estrazione delle feature si può fare una divisione tra tecniche che operano in tre domini diversi: spazio, tempo e frequenza. Per il dominio spaziale l'approccio più popolare è il Common Spatial Pattern (CSP) , con le sue varianti, a cui si vanno ad aggiungere l'Independent Component Analysis (ICA) e il filtraggio Laplaciano. Nel dominio temporale viene invece solitamente utilizzato un approccio di tipo statistico andando a valutare valore medio, deviazione standard, varianza, etc. ma anche potenza del segnale metodi e basati sui grafi. Infine nel dominio della frequenza si sfruttano la Fast Fourier Transform (FFT) la Densità Spettrale di Potenza (PSD) e la potenza della banda [29].

L'ultimo passaggio, quella della classificazione, può essere eseguito attraverso diverse tecniche di Machine Learning (ML) a seconda delle necessità e del tipo di classificazione che si intende fare (principalmente se a due classi o multiclasse). I classificatori più popolari, in particolare per le BCI basate sulla MI sono il Linear Discriminant Analysis (LDA) e il Support Vector Machine (SVM) con quest'ultimo che ha dimostrato performance migliori dell'(LDA) e del K-Nearest Neighbour (KNN) ma ha un costo computazionale maggiore. L'analisi Bayesiana inoltre ha prodotto buoni risultati al costo di un tempo di esecuzione maggiore presentando quindi un trade-off accuratezza-tempo parzialmente ridotto dall'impiego di reti neurali [17].

1.4 Problema dell'Intentional Non-Control

Con Intention Non-Control (INC) si intende quello stato in cui l'utente che utilizza la BCI non ha intenzione di impartire alcun comando.

Questo problema è stato affrontato in passato principalmente in tre modi:

(i) utilizzando una classificazione a più classi per poter modellare lo stato di

riposo, (ii) chiedendo all'utente di controllare attivamente la BCI in modo da non inviare comandi non voluti, oppure (iii) sfruttando tecniche di controllo in modo da minimizzare l'impatto di eventuali comandi involontari [30]. Queste tecniche però presentano tutte degli svantaggi: la prima è inevitabilmente affetta dalla complessità del modello per la classe di riposo; la seconda richiede un carico di lavoro troppo alto per l'utente che deve essere in controllo in ogni situazione; e infine la terza richiede un dispositivo in grado di interpretare in modo affidabile le intenzioni dell'utente a runtime.

L'idea proposta nel lavoro citato, "Do not move!" invece è quella che ci sia una correlazione tra la coerenza del segnale EEG e lo stato di INC. Nello specifico si è deciso di utilizzare l'entropia dei segnali per rilevare accuratamente questo stato.

In altri lavori [31] si è anche confrontata l'efficacia dell'utilizzo dell'entropia rispetto ad altri metodi mostrando come per esempio il contenuto informativo dell'entropia sia molto elevato quando viene eseguito un task di movimento e molto più basso invece quando l'utente è fermo.

Nel prossimo capitolo verrà approfondita più nella dettaglio la motivazione dietro la scelta dell'entropia e il suo utilizzo in questo lavoro.

1.5 Obiettivi e struttura

L'obiettivo di questo lavoro di tesi è quello di sviluppare un pacchetto ROS-Neuro [32][33] per l'elaborazione dei segnali EEG acquisiti da una BCI in grado di identificare in modo accurato e in tempo reale lo stato di INC. Di seguito è illustrata la struttura dell'elaborato.

Nel capitolo 2 "Materiali e Metodi" saranno descritti gli strumenti utilizzati e sarà spiegato l'approccio proposto con le relative motivazioni che hanno

spinto a impiegarlo. Verrà inoltre illustrato il protocollo sperimentale che si è deciso di seguire per la valutazione delle performance e il modo in cui si è ottimizzato il processo di elaborazione

Nel capitolo 3 "Risultati" saranno analizzati i risultati dell'esperimento.

Infine il capitolo 4 "Conclusioni e sviluppi futuri" conterrà l'analisi dei risultati ottenuti e la possibilità di ampliamento di questo lavoro.

2 | Materiali e Metodi

Come menzionato in "Obiettivi e struttura" (1.5) in questo capitolo saranno descritti gli strumenti, le librerie e i dispositivi utilizzati per la realizzazione dell'interfaccia.

2.1 Ambiente di sviluppo

2.1.1 Robot Operating System

Il Robot Operating System, conosciuto più semplicemente come ROS ¹, è un insieme di framework open-source per lo sviluppo e la programmazione di robot. Fornisce astrazione dell'hardware, driver dei dispositivi, librerie, strumenti di visualizzazione, la gestione dei package, l'implementazione di funzionalità usate spesso e lo scambio dei messaggi tra i vari processi. In questo lavoro è stata usata la versione ROS Noetic.

Questo sistema operativo si è diffuso largamente grazie al suo modo di gestire le molte parti che compongono il comportamento finale di un robot e questo risultato è stato ottenuto grazie alla sua programmazione modulare e alla sua natura open-source. È possibile sviluppare utilizzando i più comuni linguaggi di programmazione come C++ e Python.

¹<https://www.ros.org>

ROS si compone di tre livelli che sono: Filesystem Level, Computation Graph Level e Community Level.

Filesystem Level

Questo livello comprende le risorse ROS che sono presenti su disco come:

- **Package:** sono l'unità principale per l'organizzazione del software in ROS. Può presentare al suo interno più processi, o nodi, librerie e file di configurazione.
- **Metapackage:** sono package specializzati utilizzati per rappresentare un gruppo di package legati tra loro.
- **Package Manifest:** forniscono i metadati del package.
- **Repository:** un insieme di package condivisi attraverso un comune sistema di controllo versione (Git, SVN, etc.).
- **Messaggi:** definiscono la struttura dati di un messaggio inviato in ROS.
- **Servizi:** definiscono la struttura dati per una richiesta e risposta per servizi in ROS.

Computation Graph Level

Questo livello rappresenta il network peer-to-peer dei processi ROS che elaborano insieme i dati e comprende:

- **Nodi:** processi che eseguono la computazione, spesso un sistema di controllo robotico prevede più nodi.

- Master: fornisce la registrazione dei nomi e l'accesso al resto del livello di Computation Graph, senza la presenza del master i nodi non sarebbero in grado di trovarsi tra di loro.
- Parameter Server: è parte del Master e permette di memorizzare i dati in una posizione centrale.
- Topic: i messaggi sono indirizzati ai vari nodi tramite un sistema publish/subscribe. Un nodo invia un messaggio pubblicando su un dato topic e questo poi potrà essere letto dai nodi che eseguiranno la subscribe sullo stesso topic.
- Bags: sono un formato per salvare e recuperare dati di messaggi ROS, sono molto utili per recuperare informazioni difficili da collezionare ma fondamentali per lo sviluppo come per esempio dati di sensori.

Community Level

Quest'ultimo livello è composto da tutte quelle risorse che permettono a community diverse di scambiarsi conoscenza e software, quali ad esempio packages e repositories. Istituzioni o singoli individui possono sviluppare e rilasciare una propria componente rendendola disponibile a tutti. Altre risorse sono inoltre ROS Wiki, che fornisce materiali come tutorial per installazione e per muovere i primi passi con ROS, e ROS Answer, una sezione del sito che permette agli utenti di fare domande relative a ROS ricevendo il supporto della community.

2.1.2 ROS-Neuro

ROS-Neuro² è un middleware open-source basato su ROS [32][33] che nasce per rendere disponibile un ecosistema comune per lo sviluppo di applicativi basati sull'interazione tra robotica e neuroscienze.

I principali moduli disponibili in ROS-Neuro sono i seguenti:

- `rosneuro_acquisition`: gestisce l'acquisizione dei segnali neurali da una serie diversa di dispositivi e li pubblica su uno specifico topic sottoforma di messaggio personalizzato di tipo `NeuroFrame`.
- `rosneuro_recorder`: permette di registrare e salvare su file i dati acquisiti
- `rosneuro_processing`: elabora il messaggio `NeuroFrame` ricevuto e pubblica come messaggio `NeuroOutput` l'intenzione decodificata. In questo lavoro di tesi si è proceduto a creare un pacchetto di questo tipo.
- `rosneuro_feedback`: implementa un'interfaccia grafica che fornisce un feedback all'utente su quelle che sono le azioni decodificate.
- `rosneuro_decisionmaking`: implementa algoritmi di decision making che interpretano il messaggio `NeuroOutput` in base ai requisiti della specifica applicazione.
- `rosneuro_control`: traduce il messaggio `NeuroOutput` in un segnale di controllo.

²<https://github.com/rosneuro>

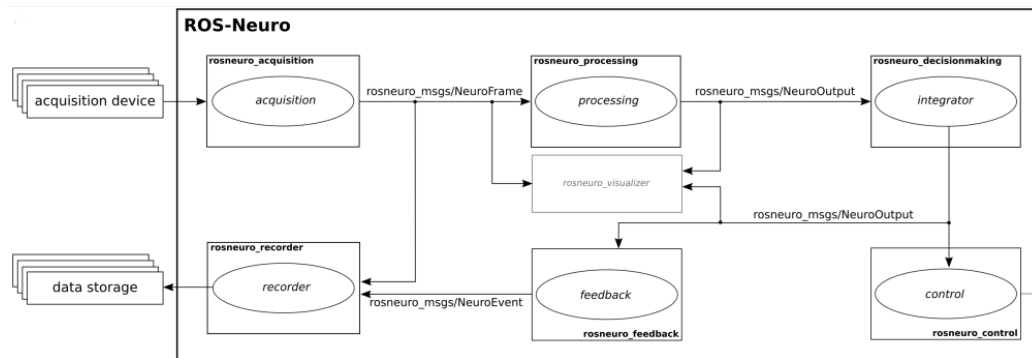


Figura 2.1: Rappresentazione schematica dell'architettura di ROS-Neuro

2.2 Scelta dell'entropia

Come accennato in "Problema dell'Intentional Non-Control" (1.4) l'entropia dei segnali EEG si è dimostrata efficace per affrontare il problema dell'INC. In particolare questo è stato poi verificato successivamente da altri studi e lavori.

Si è deciso di prendere in considerazione come riferimento e punto di partenza il lavoro "Entropy-based Motion Intention Identification for Brain-Computer Interface" [34], in cui l'entropia viene utilizzata proprio per il rilevamento dell'Intentional Non-Control in un sistema con una BCI controllata tramite MI. Il preprocessing dei segnali prevedeva un filtro notch a 50 Hz per rimuovere il rumore generato dall'amplificazione e un filtro spaziale di tipo CAR. L'analisi sull'entropia invece è stata effettuata dividendo i dati in finestre da 1,5 secondi con sovrapposizioni di 0,125 secondi, risultate in un framerate di 8 Hz. Sono poi state estratte sei bande di frequenza utilizzando un filtro di Butterworth di ordine quattro e per ognuna di queste bande è stato calcolato l'involuppo della trasformata di Hilbert del segnale. Infine a questo punto è stata calcolata l'entropia per ogni banda. Per segnali continui questa è

definita come

$$H_{sh} = - \sum_i P_i \log P_i$$

dove i varia su tutto il range di ampiezza del segnale e P_i indica la probabilità che il segnale abbia ampiezza α_i .

Per un segnale continuo invece si può calcolare come la media di istogrammi, dividendo linearmente il range di ampiezza del segnale in k bins. Infine viene usata l'entropia di Shannon normalizzata calcolata come

$$nH_{Sh} = \frac{H_{Sh}}{\log k}$$

Nello specifico in questo caso sono stati utilizzati 32 bins.

Le features più significative sono state scelte utilizzando la Canonical Variate Analysis per distinguere tra i due stati (INC e IC), ed usata poi per l'estrazione dei Canonical Discriminant Spatial Patterns. L'esperimento consisteva poi in quattro fasi consecutive, guidate tramite un'interfaccia grafica mostrata in Figura 2.2, in cui al soggetto viene chiesto di "sparare" verso un rettangolo presente a schermo. Le fasi sono:

- INC-stationary: non ci sono obiettivi presenti a schermo e al soggetto viene chiesto di rimanere fermo;
- IC-cue: a schermo inizia ad apparire un avvertimento per il soggetto che gli indica che a breve dovrà eseguire un'azione;
- IC-stationary: il soggetto deve eseguire l'azione aprendo e chiudendo le mani in modo continuo;
- INC-cue: il bordo destro del rettangolo appare a schermo, il soggetto deve continuare la sua azione ma sa che a breve dovrà interromperla.

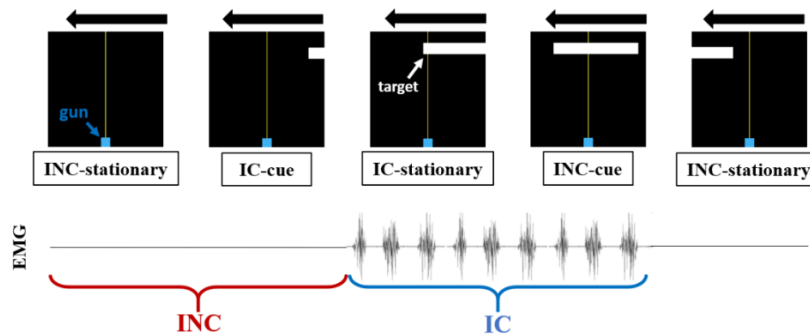


Figura 2.2: Interfaccia mostrata all'utente nell'esperimento

Dai risultati di questo lavoro si può evincere come un'alto valore dell'entropia indichi una maggiore attenzione e comandi recapitati più velocemente alla BCI e bassa entropia un invio più lento dovuto alla bassa attenzione. I risultati di questo esperimento hanno confermato come l'entropia possa rappresentare uno strumento prezioso per affrontare il problema dell'INC e in questo lavoro di tesi si è infatti deciso di seguire un approccio simile a quanto descritto sopra.

2.3 Modello proposto

Per la realizzazione del package ROS-Neuro si è scelto di utilizzare il linguaggio Python e in questa sezione si descriverà la sua struttura, quali sono state le scelte per l'elaborazione dei segnali EEG acquisiti e come si è strutturato l'esperimento.

Il package, chiamato *rosneuro_processing_entropy*, si occuperà dell'elaborazione dei segnali acquisiti tramite il *rosneuro_acquisition* e dopo tale operazione verrà effettuata la classificazione e quindi la decodifica dell'azione eseguita dall'utente.

Secondo quanto detto nella sezione precedente il modello proposto prende

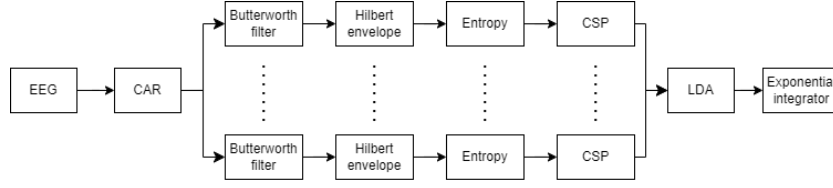


Figura 2.3: Rappresentazione schematica dell'approccio usato per l'elaborazione e la classificazione dei segnali. Le ramificazioni indicano le diverse bande utilizzabili.

come punto di partenza il lavoro esaminato e descritto sopra e altri studi simili eseguiti successivamente [35]. Dalla figura 2.3 si può avere una visione d'insieme del suddetto modello i cui passi saranno adesso descritti più nello specifico.

Il primo passo una volta acquisiti i segnali consiste nel pre-processing per rimozione del rumore e miglioramento del segnale stesso, che viene effettuato tramite un'implementazione di un filtro CAR. A questo punto i dati vengono filtrati con più filtri Butterworth, su bande di frequenza diverse, e si è scelto di utilizzare filtri di ordine quattro e su ogni banda viene prima calcolato l'involuppo della trasformata di Hilbert del segnale e infine l'entropia di Shannon.

Per la classificazione e quindi per la determinazione delle intenzioni dell'utente vengono prima estratte le features più significative tramite un CSP e queste vengono date in input a un classificatore di tipo LDA. Prima dell'output effettivo le predizioni del classificatore passano attraverso un integratore esponenziale di questo tipo

$$D(y_t) = \alpha \cdot D(y_{t-1}) + (1 - \alpha) \cdot p(y_t|x_t)$$

dove $D(y_t)$ è la probabilità integrata a posteriori della classe y e $p(y_t|x_t)$ è la probabilità a posteriori fornita dal classificatore con input x_t e α è un parametro di integrazione, fissato a 0.9 in questo caso.

2.4 Interfaccia di training offline

Per poter effettuare il training del classificatore è stata fornita un'interfaccia grafica (Figura 2.4) che presenta le seguenti funzionalità:

- Impostazioni su buffer, lunghezza della finestra, framerate per la classificazione (windows shift) e numero di canali utilizzati in acquisizione;
- Impostazione dei filtri, permettendo di scegliere l'ordine e la frequenza delle bande;
- Impostazioni per la classificazione dove vengono indicati i bins per il calcolo dell'entropia e la dimensione dei CSP (e cioè le features estratte);
- Impostazioni dell'integratore esponenziale;
- Selezione del tipo di pre-processing desiderato tra CAR e filtro Laplaciano;
- Selezione dei file .gdf contenenti i dati sulle acquisizioni
- Selezione dei codici per le classi (in questo lavoro 770 per l'INC, 771 per l'immaginazione del movimento delle mani e 773 per l'immaginazione del movimento dei piedi)

Classifier Training

Settings

Buffer Settings

Sampling Rate: 512

Window Length: 1.5

Window Shift: 0.0625

Num Chans: 32

Filter Settings

Filter Order: 4

Lower cut-off frequency: 8

Higher cut-off frequency: 35

Order	Lower cut-off	Higher cut-off
4	16.0	30.0
4	2.0	45.0
4	8.0	13.0
4	8.0	35.0

Classification Settings

Entropy Bins Number: 128

CSP Dimension: 8

Integrator Settings

Alpha: 0.9

Integrator Threshold: 0.5

Integrator Rejection: 0.5

Integrator Begin: 0

Training Settings

☒ Save Classifier

Classifier ID:

Preprocessing Options

☐ Laplacian ☒ CAR

Select file No file selected.

Restore Default Settings

Class Selection

Load gdf

/home/diego/d1.20230215.153

First Class Codes:

0 Add

☐ 771

☐ 773

☒ 770

Second Class Codes:

0 Add

☒ 771

☒ 773

☐ 770

GUI Options

☐ Load Data

☐ Log data

Launch Training

Figura 2.4: Applicativo utilizzato per il training del classificatore

Dopo l'elaborazione dei dati acquisiti, che segue lo stesso modello descritto in precedenza, viene presentata una schermata di selezione delle features (Figura 2.5), dove per ogni banda è riportato il Fisher score [36] associato alle features.

Una volta inserite le varie impostazioni viene eseguito il training del clas-

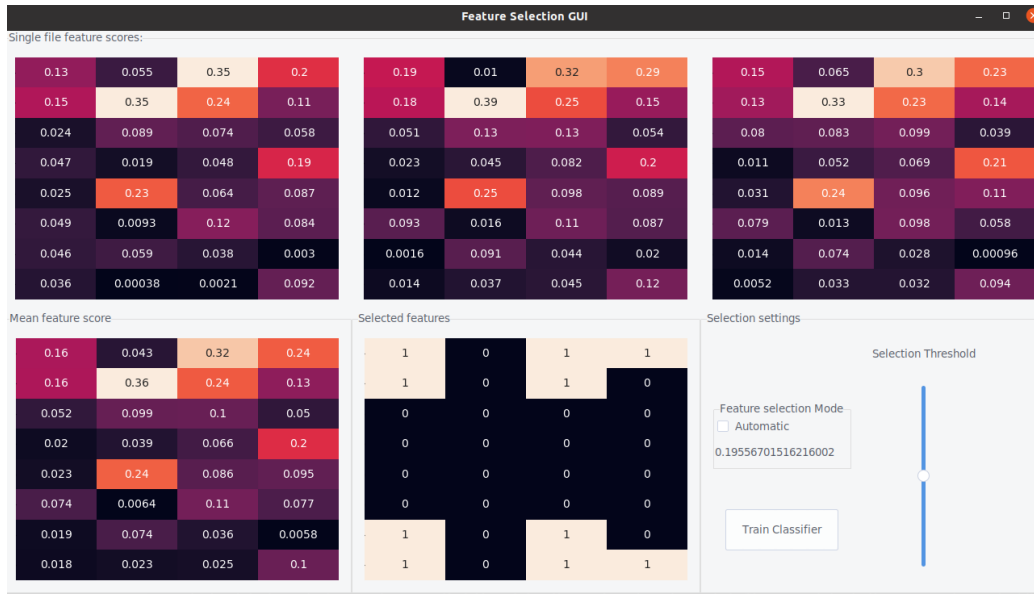


Figura 2.5: Schermata di selezione delle features. In alto i Fisher score calcolati sui singoli file caricati; in basso a sinistra il Fisher score medio; in basso al centro le features selezionate.

sificatore e viene restituito un file di configurazione contenente tutte le specifiche indicate e il classificatore stesso che potrà essere utilizzato per le acquisizioni in tempo reale.

2.5 Test e ottimizzazione

Una volta terminato lo sviluppo, prima di passare all'esperimento vero e proprio, si è proceduto a testare il package per verificare sia i tempi di esecuzione (in modo da poter scegliere in modo adeguato le finestre temporali e il framerate per la classificazione) sia per essere in grado di scegliere adeguatamente gli altri parametri. Per i test è stato utilizzato un file .gdf di acquisizione a 16 canali.

La prima valutazione è stata quella riguardo ai tempi di esecuzione: si è

verificato che in tempo reale il sistema fosse in grado di effettuare la classificazione e la predizione delle intenzioni dell'utente in una finestra temporale di 62,5 ms (framerate 16 Hz) senza ritardi. Dai primi test però, utilizzando più di tre bande di frequenza per il filtraggio, questi tempi venivano superati e si è quindi proceduto ad analizzare il processo in modo da individuare quali operazioni presentassero una complessità maggiore.

La prima modifica ha interessato la libreria fornita per il calcolo dell'entropia: il calcolo degli istogrammi implementato aveva una complessità temporale $O(n^2)$ e si è riusciti ad abbassare questa complessità a $O(n)$ sfruttando alcune proprietà della libreria NumPy³, evitando quindi di dover eseguire un ciclo di iterazioni sui vettori coinvolti nei metodi. Nonostante ciò i tempi con l'utilizzo di 4 bande sono rimasti elevati e si è quindi deciso di optare per un downsampling per ridurre ulteriormente il tempo impiegato per il calcolo dell'entropia.

Un'ultima modifica si è resa necessaria prima di poter procedere all'esperimento vero e proprio: un meccanismo di sicurezza nel caso in cui la qualità dei dati acquisiti non sia abbastanza buona. Questa criticità si è presentata durante alcuni degli ultimi test durante l'estrazione delle features con il CSP. Per poter essere effettuata questa operazione è necessario che le colonne della matrice *campioni x canali* in input al CSP abbia tutte le colonne (canali) linearmente indipendenti. Nel caso in cui i dati non siano di buona qualità si può verificare il caso in cui due o più colonne siano linearmente dipendenti e quindi la procedura vada in errore. Per ovviare a questo problema è stato fatto in modo che in questa circostanza, prima del calcolo del CSP vengano estratte dalla matrice le sole colonne indipendenti e qualora queste dovessero essere inferiori alla dimensione indicata per il CSP

³<https://numpy.org/>

quest'ultima verrebbe ridotta al suddetto numero di colonne linearmente indipendenti.

Una volta terminati i test si è quindi potuto procedere all'esperimento.

2.6 Struttura dell'esperimento

L'esperimento si divide in due fasi:

- **Traning offline:** l'utente segue le istruzioni a schermo, fornite tramite il package *rosneuro_feedback*, eseguendo i tre task associati (Figura 2.6) e cioè immaginazione del movimento delle mani, immaginazione del movimento dei piedi e INC per un periodo di tempo casuale tra 3.5 e 4.5 secondi, determinato dal riempimento di una delle tre barre verticali. Sono state eseguite tre run da 45 trials l'una (15 trials per ogni classe).
- **Acquisizione online:** sempre tramite il package *rosneuro_feedback* l'utente, in modo simile alla fase precedente, segue le istruzioni a schermo, ma in questo caso il riempimento della barre verticali è legato all'esecuzione del task, che sarà considerato terminato una volta riempita la barra in questione.

Nella fase di training l'utente esegue per 15 volte ogni task, in ordine casuale, e questa operazione viene ripetuta per tre volte generando quindi tre file .gdf che verranno usati per il training vero e proprio del classificatore sfruttando l'interfaccia descritta in precedenza

Sono state scelte quattro bande di frequenza per i filtri Butterworth di ordine quattro: 16-30 Hz, 2-45 Hz, 8-13 Hz e 8-35 Hz.

Sulla base delle considerazioni fatte nella sezione 2.5 si è scelto di calcolare l'entropia utilizzando 128 bins.

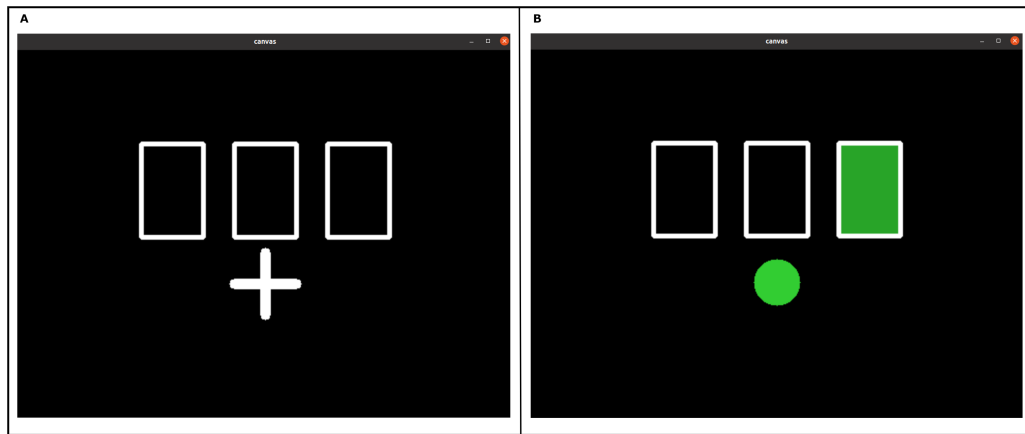


Figura 2.6: (A): fase di fixation in cui l'utente viene avvertito che sta per arrivare un comando; (B): fase di riempimento della barra in cui l'utente esegue il comando indicato

La dimensione del CSP è di 8 features per ogni banda (le 4 più significative per ogni classe), di cui vengono poi selezionate le migliori sulla base del Fisher Score.

Nella fase online viene letto il file contenente tutti i dati di configurazione utilizzate nella fase precedente e i blocchi di dati ricevuti dal dispositivo di acquisizione vengono classificati a un framerate di 16 Hz.

2.6.1 Dispositivo di acquisizione EEG

Per l'acquisizione dei segnali elettroencefalografici è stata usata la g.USBamp⁴. Si tratta di un amplificatore di biosegnali ad alta accuratezza e alte performance con un sistema di acquisizione ed elaborazione.

Il sistema include amplificatori, software di acquisizione e di analisi e supporta 16 canali campionati contemporaneamente a 24 bit di risoluzione. Un totale di 4 messe a terra indipendenti garantisce che non ci siano interferenze

⁴<https://www.gtec.at/product/gusbamp-research/>

tra i segnali registrati. L'amplificatore si può collegare facilmente a una porta USB del PC ed essere utilizzato immediatamente per la registrazione dei dati. La g.USBamp ha un range di input di ± 250 mV, che permette la registrazione di segnali DC senza saturazione.

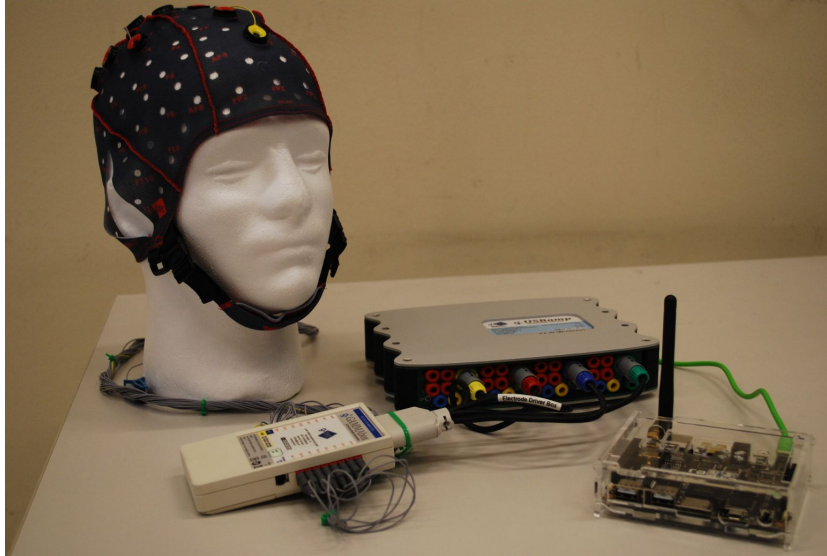


Figura 2.7: Dispositivo di acquisizione EEG gUSBamp

3 | Risultati

In questo capitolo saranno riportati i risultati ottenuti nella prima fase di test e ottimizzazione e nella procedura sperimentale, sia nella prima fase di training e creazione del classificatore, sia nella seconda di esecuzione in tempo reale. Per ciò che concerne l'esperimento si prenderà in considerazione solo l'entropia nelle bande selezionate (in questo caso solo la banda 8-13 Hz).

Le features migliori durante la fase di training sono risultate essere quelle relative alla banda 8-13 Hz, in particolare il Fischer score delle features associate alla prima classe (INC) è stato significativamente più elevato rispetto a quello delle features della seconda classe (movimento) (Figura 3.1).

Il classificatore così ottenuto ha $accuracy = 0.8453$ e la seguente matrice di confusione

$$\begin{bmatrix} 382 & 77 \\ 65 & 394 \end{bmatrix}$$

Si è proceduto quindi poi con l'esperimento in tempo reale e in questo caso i risultati, in termini di HIT e MISS del classificatore, sono stati i seguenti: su 15 trial di INC 14 volte è stata riconosciuta correttamente la classe mentre su 30 trial di movimento (15 per le mani e 15 per i piedi) la classe è stata riconosciuta correttamente solo una volta.

Al fine di capire anche meglio questi risultati si è provveduto ad analizzare

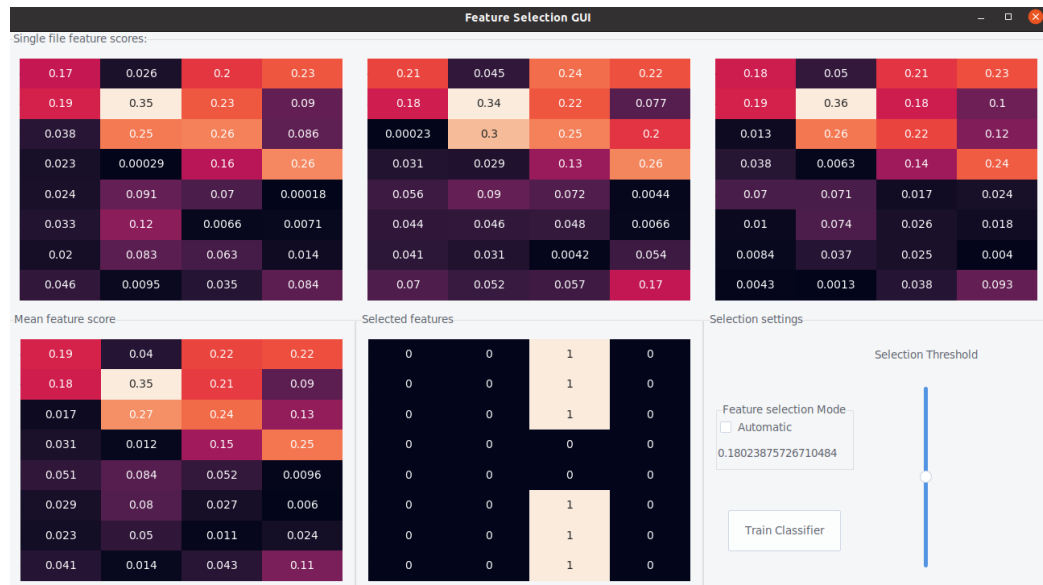


Figura 3.1: Punteggi delle features dopo aver eseguito le acquisizioni di calibrazione. In ordine da sinistra a destra i punteggi della features (in verticale) per banda: 16-30 Hz, 2-45 Hz, 8-13 Hz, 8-35 Hz. In basso al centro è possibile vedere quali sono le features selezionate.

nel dettaglio i dati acquisiti in entrambe le fasi.

3.1 Output dei test

Per valutare le performance del downsampling si è prima proceduto a esaminare le curve dell'entropia calcolata normalmente e quella ottenuta tramite downsampling come mostrato in Figura 3.2 (per praticità viene raffigurato uno solo dei 16 canali).

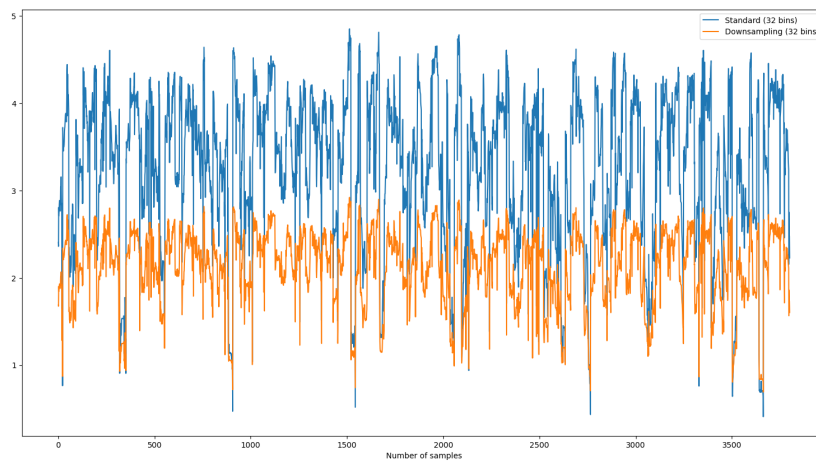


Figura 3.2: Confronto tra entropia calcolata in modo standard e tramite downsampling a 32 bins

Come si può notare però questo approccio taglia in modo netto i valori più alti dell'entropia a causa del numero di bins mantenuto uguale, per questo motivo si è ripetuto questo test con altri valori di bins superiori per il downsampling (Figure 3.3, 3.4, 3.5, 3.6, 3.7)

Con l'aumento del numero dei bins per gli istogrammi, in particolare per 128, 192, 256 e 512 bins, la differenza tra le curve si è ridotta e quindi si è provveduto ad analizzare le performance utilizzando questo metodo (Figura 3.8). Come si può notare tutti i quattro casi che sfruttano il downsampling

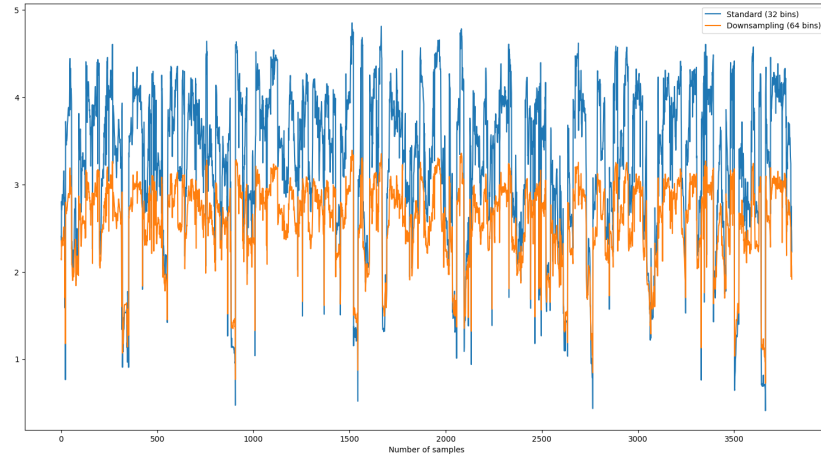


Figura 3.3: Confronto tra entropia calcolata in modo standard e tramite downsampling a 64 bins

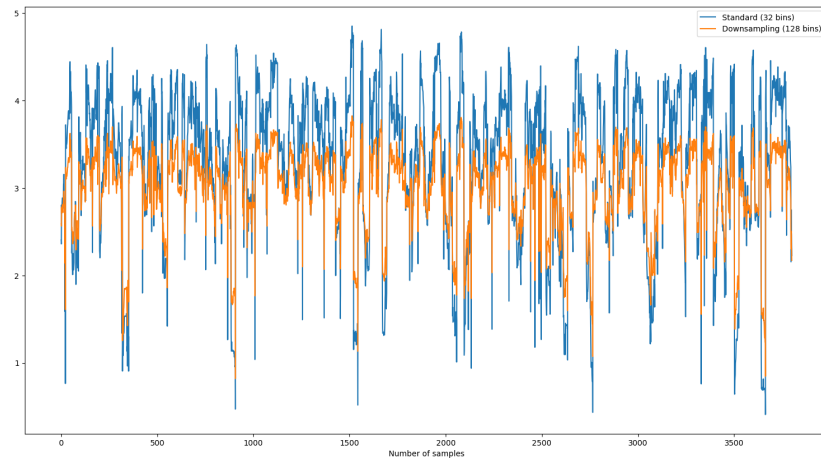


Figura 3.4: Confronto tra entropia calcolata in modo standard e tramite downsampling a 128 bins

mostrano un miglioramento significativo in termini di tempo di esecuzione rendendo quindi necessaria un'ulteriore analisi per determinare quale delle

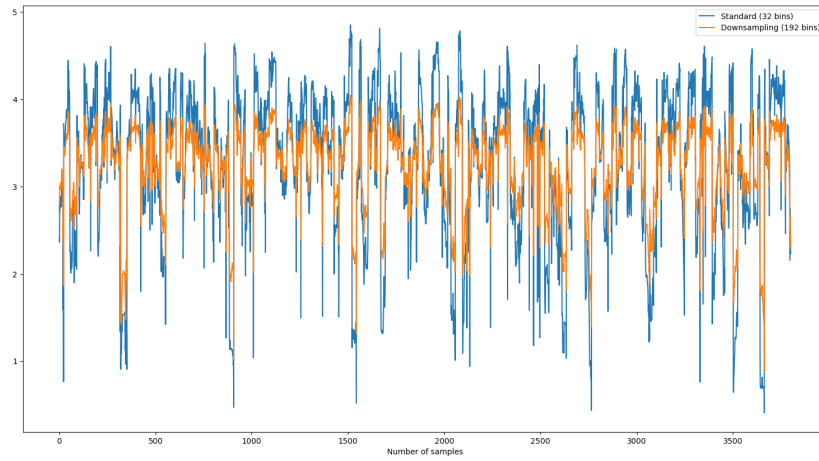


Figura 3.5: Confronto tra entropia calcolata in modo standard e tramite downsampling 192 bins

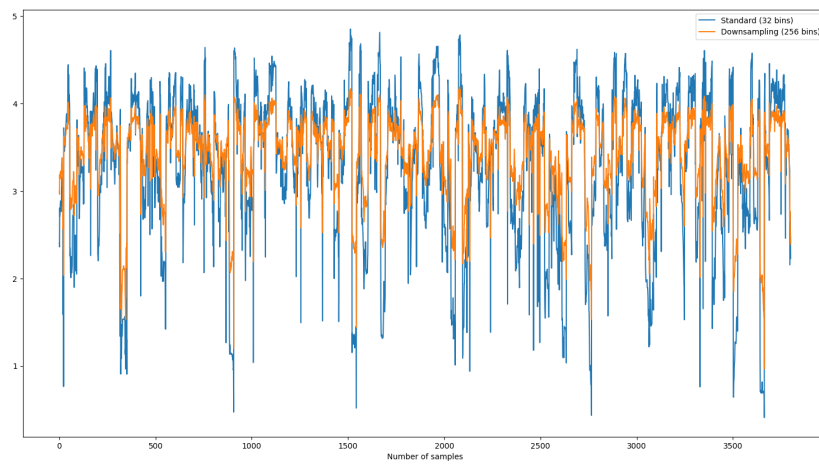


Figura 3.6: Confronto tra entropia calcolata in modo standard e tramite downsampling a 256 bins

quattro opzioni fosse la più adatta. Per fare ciò si è quindi deciso di valutare la somiglianza tra gli istogrammi, in termini di curve Gaussiane che li ap-

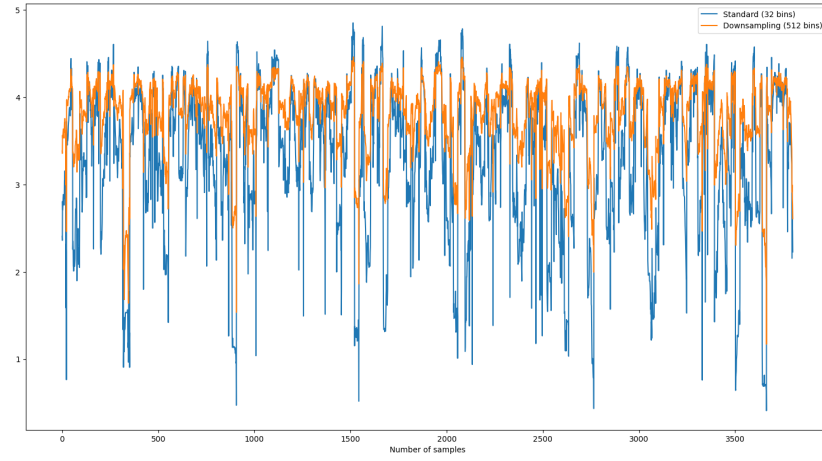


Figura 3.7: Confronto tra entropia calcolata in modo standard e tramite downsampling a 512 bins

prossimano, sulla base del test di Kolmogorov-Smirnov [37].

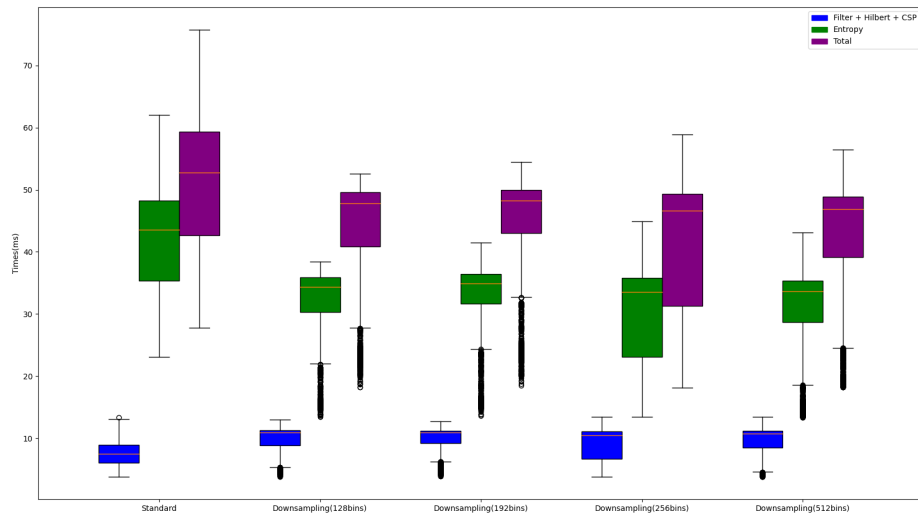


Figura 3.8: Confronto tra i tempi di esecuzione tra i vari metodi usati per il calcolo dell'entropia

Come si può vedere in Figura 3.9 la curva "meno diversa" rispetto all'originale è quella ottenuta con il downsampling a 128 bins.

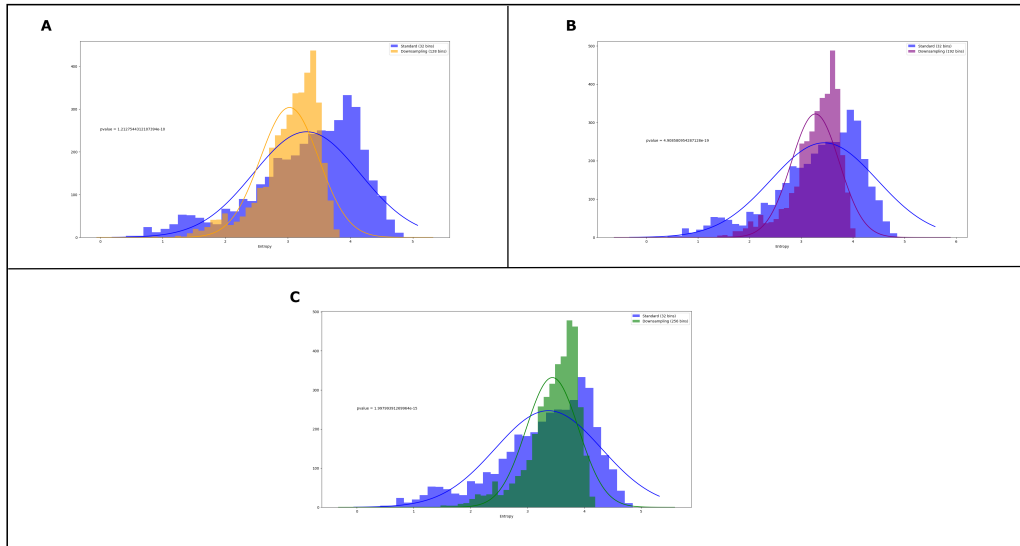


Figura 3.9: Confronto tra gli istogrammi con l'uso del test di Kolmogorov-Smirnov. (A: 128 bins, B: 192 bins, C: 256 bins)

3.2 Risultati offline

Per valutare le acquisizioni si è provveduto come prima cosa a calcolare il valore medio e la deviazione standard dell'entropia per ognuna delle tre classi per ogni elettrodo (canale) e valore medio e deviazione standard del CSP.

È subito possibile notare come durante i trial di INC l'entropia abbia dei valori significativamente più bassi rispetto a quella dei trial di movimento, come più facilmente osservabile in tabella 3.1.

In figura 3.11 sono riportati gli andamenti dell'entropia media durante i vari trial, divisi per canale. Anche qui si può notare che l'entropia associata ai

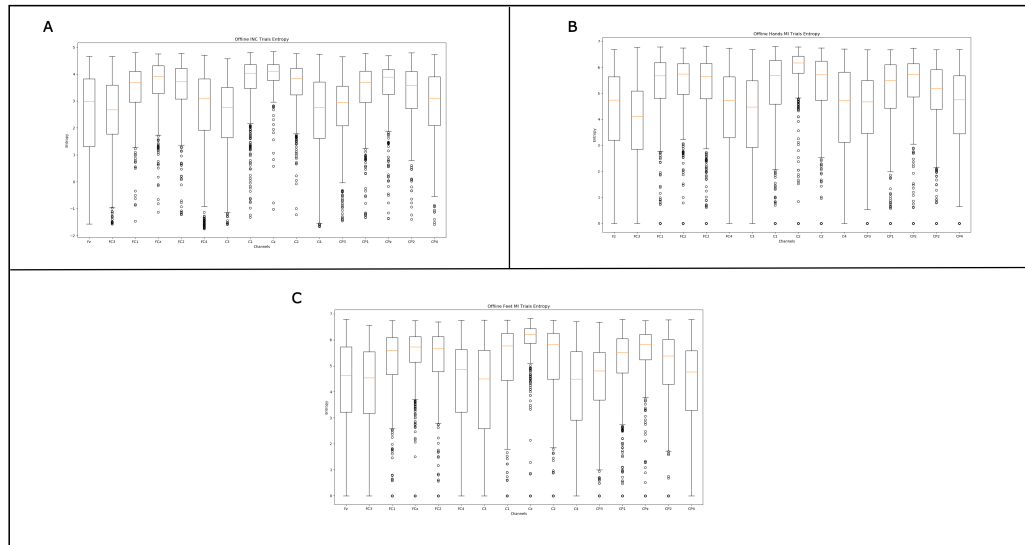


Figura 3.10: A: Valore medio e deviazione standard dell'entropia calcolata durante l'INC offline; B: Valore medio e deviazione standard dell'entropia calcolata durante la motor imagery delle mani offline; C: Valore medio e deviazione standard dell'entropia calcolata durante la motor imagery dei piedi offline

trial di INC è più bassa degli altri due in quasi ogni canale. Il primo secondo del grafico (da -1 a 0) è il tempo in cui è presente a schermo l'indicazione sul comando da eseguire, il restante è invece l'esecuzione del task da parte dell'utente.

Questo conferma l'ipotesi di partenza secondo cui all'INC sono associati valori più bassi di entropia rispetto ai task di movimento confermando la validità dell'approccio scelto.

Le stesse analisi riportate sopra sono state anche effettuate moltiplicando la matrice dell'entropia per quella dei coefficienti del CSP (ottenuta durante il training). Di seguito sono mostrati i relativi grafici: prima mostrando i valori medi per le tre classi (figure ??, ??, ??) e poi i valori medi e le deviazioni standard durante i vari trial (figura 3.13)

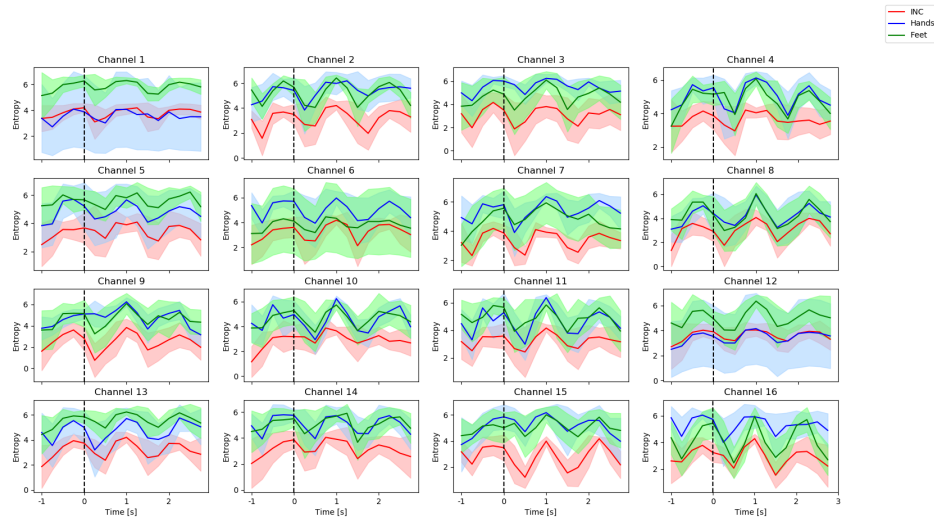


Figura 3.11: Oscillazione dell'entropia media durante un trial per ogni canale durante l'acquisizione offline.

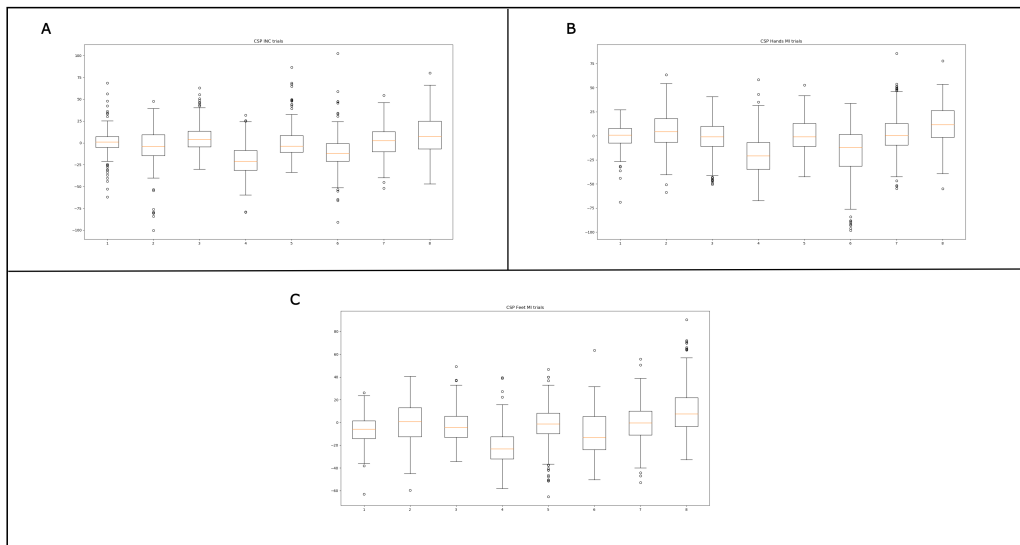


Figura 3.12: A: Valore medio e deviazione standard del CSP calcolato durante l'INC offline; B: Valore medio e deviazione standard del CSP calcolato durante la motor imagery delle mani offline; C: Valore medio e deviazione standard del CSP calcolato durante la motor imagery dei piedi offline.

	INC	MI Mani	MI Piedi
Canale Fz	2.45 ± 1.63	4.28 ± 1.67	4.28 ± 1.70
Canale FC3	2.53 ± 1.33	3.86 ± 1.54	4.18 ± 1.57
Canale FC1	3.44 ± 0.92	5.20 ± 1.47	5.18 ± 1.33
Canale FCz	3.70 ± 0.88	5.30 ± 1.39	5.45 ± 1.12
Canale FC2	3.49 ± 0.98	5.20 ± 1.50	5.30 ± 1.26
Canale FC4	2.62 ± 1.61	4.21 ± 1.87	4.33 ± 1.77
Canale C3	2.48 ± 1.34	4.07 ± 1.67	4.17 ± 1.68
Canale C1	3.72 ± 1.00	5.17 ± 1.57	5.24 ± 1.43
Canale Cz	4.01 ± 0.54	5.78 ± 1.34	5.96 ± 1.03
Canale C2	3.58 ± 0.88	5.24 ± 1.48	5.27 ± 1.37
Canale C4	2.55 ± 1.48	4.26 ± 1.82	4.18 ± 1.68
Canale CP3	2.70 ± 1.22	4.30 ± 1.60	4.89 ± 1.38
Canale CP1	3.40 ± 1.04	4.99 ± 1.55	5.13 ± 1.34
Canale CPz	3.59 ± 0.90	5.26 ± 1.42	5.21 ± 1.14
Canale CP2	3.30 ± 1.04	4.88 ± 1.47	4.98 ± 1.41
Canale CP4	2.58 ± 1.27	4.39 ± 1.61	4.38 ± 1.51

Tabella 3.1: Valore medio e deviazione standard dell'entropia per le acquisizioni offline.

In quest'ultima figura si può notare come i pattern siano più sovrapposti.

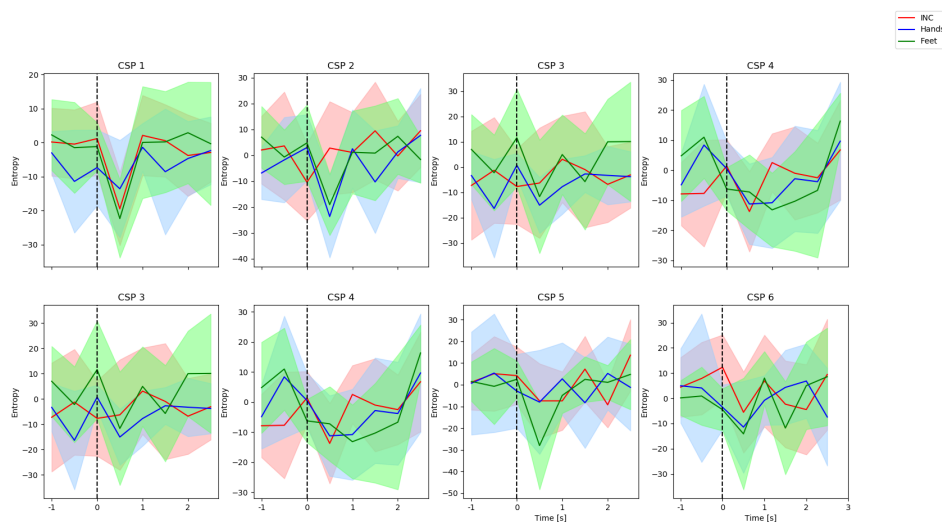


Figura 3.13: Oscillazione dei valori medi del CSP calcolata durante i trials offline.

3.3 Risultati online

In questa sezione saranno invece riportati i risultati ottenuti durante l'esecuzione in tempo reale in modo analogo a quanto fatto nella sezione precedente.

Sia dalla figura 3.14 che dalla tabella 3.2 si può notare come la differenza tra l'entropia dell'INC e quella dei task di movimento sia visibilmente minore, con la prima che è mediamente più alta mentre la seconda è mediamente più bassa.

Tale risultato si può osservare guardando anche in questo caso l'andamento medio per trial dell'entropia (figura 3.16)

In particolare in quest'ultima figura si nota come le curve dell'entropia siano molto simili rendendo quindi difficile la corretta classificazione. In

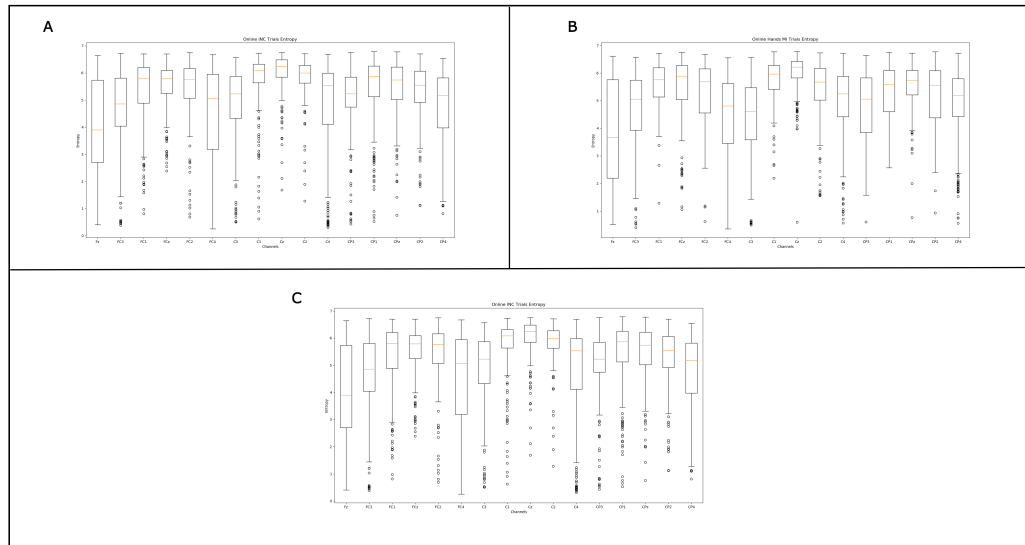


Figura 3.14: A: Valore medio e deviazione standard dell'entropia calcolata durante l'INC online; B: Valore medio e deviazione standard dell'entropia calcolata durante la motor imagery delle mani online; C: Valore medio e deviazione standard dell'entropia calcolata durante la motor imagery dei piedi online.

alcuni casi l'entropia associata ai task di INC supera anche quella osservata durante i task di movimento.

	INC	MI Mani	MI Piedi
Canale Fz	4.06 ± 1.69	3.83 ± 1.82	4.12 ± 1.72
Canale FC3	5.66 ± 1.44	4.57 ± 1.50	4.75 ± 1.30
Canale FC1	5.38 ± 1.25	5.56 ± 0.82	5.76 ± 0.93
Canale FCz	5.56 ± 0.83	5.50 ± 1.11	5.93 ± 0.58
Canale FC2	5.48 ± 1.05	5.26 ± 1.14	5.37 ± 1.34
Canale FC4	4.38 ± 1.96	4.41 ± 1.51	4.62 ± 1.61
Canale C3	4.86 ± 1.37	4.37 ± 1.43	4.58 ± 1.28
Canale C1	5.75 ± 1.03	5.77 ± 0.74	5.93 ± 0.85
Canale Cz	6.04 ± 0.73	6.01 ± 0.65	5.99 ± 0.72
Canale C2	5.87 ± 0.70	5.36 ± 1.09	5.79 ± 0.69
Canale C4	4.82 ± 1.69	4.88 ± 1.41	5.30 ± 1.20
Canale CP3	5.07 ± 1.18	4.79 ± 1.28	4.63 ± 1.23
Canale CP1	5.44 ± 1.25	5.35 ± 0.89	5.34 ± 0.98
Canale CPz	5.48 ± 1.03	5.54 ± 0.81	5.56 ± 0.92
Canale CP2	5.33 ± 1.02	5.14 ± 1.17	5.28 ± 1.02
Canale CP4	4.78 ± 1.28	4.83 ± 1.41	5.04 ± 1.26

Tabella 3.2: Valore medio e deviazione standard dell'entropia per le acquisizioni online.

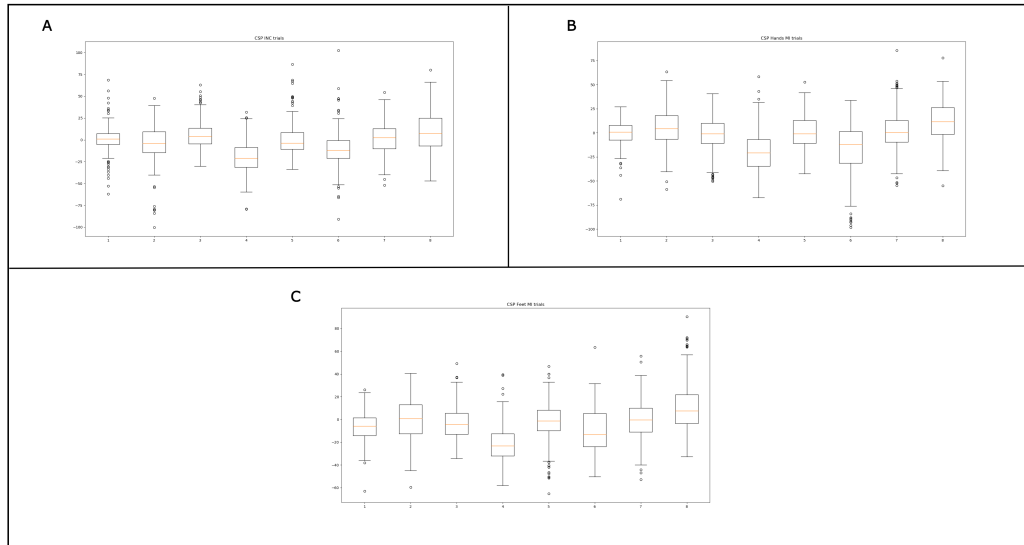


Figura 3.15: A: Valore medio e deviazione standard del CSP calcolato durante l'INC online; B: Valore medio e deviazione standard del CSP calcolato durante la motor imagery delle mani online; C: Valore medio e deviazione standard del CSP calcolato durante la motor imagery dei piedi online.

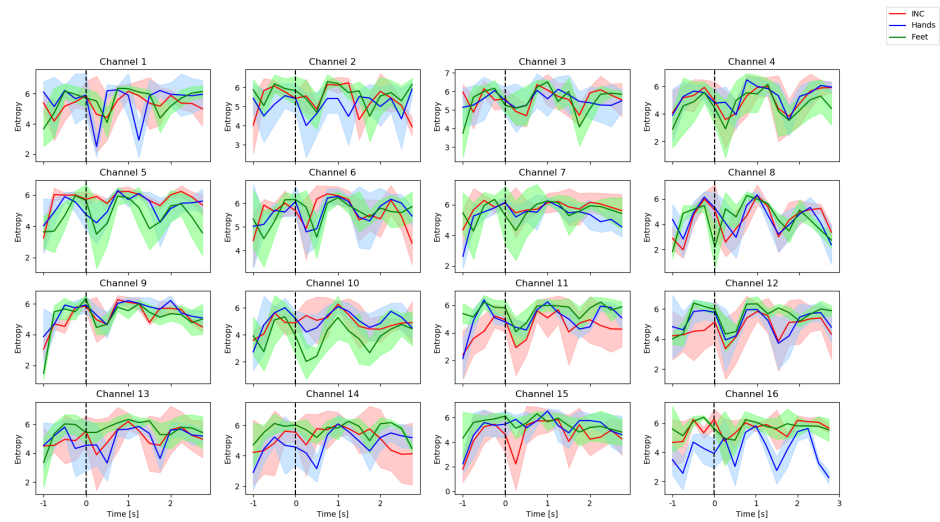


Figura 3.16: Oscillazione dell'entropia media durante un trial per ogni canale durante l'acquisizione in tempo reale.

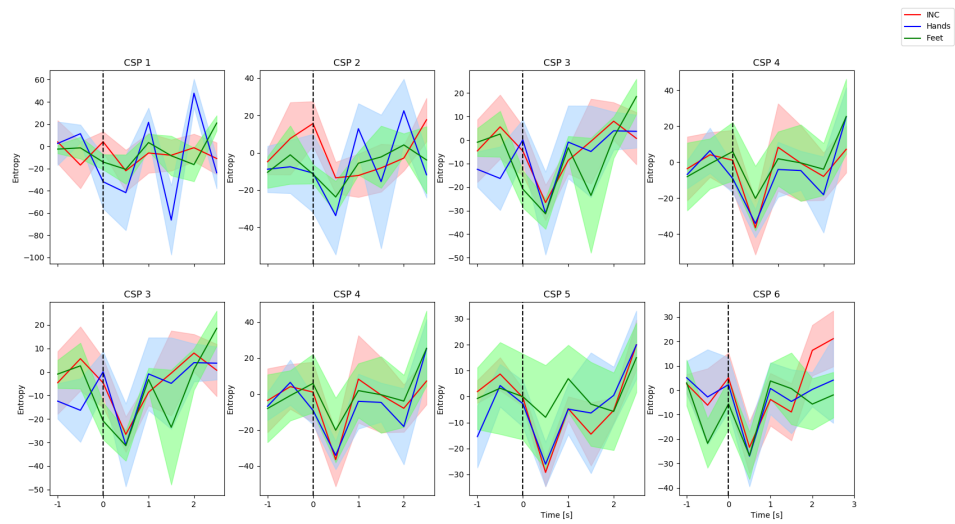


Figura 3.17: Oscillazione dei valori medi del CSP calcolata durante i trials online.

4 | Conclusioni

4.1 Discussione dei risultati

I risultati ottenuti in questo lavoro nonostante non siano ottimali sono comunque incoraggianti e possono rappresentare un passo in avanti nell'affrontare il problema dell'INC. In particolare come mostrato in "Risultati offline" l'entropia può essere uno strumento valido per la distinzione tra lo stato di movimento e quello di non-controllo in quanto l'entropia dei segnali EEG acquisiti durante un'attività di immaginazione del movimento (sia delle mani che dei piedi) ha valori notevolmente più elevati rispetto all'altra, questo è stato anche confermato dalle statistiche del classificatore ottenuto che presenta un'accuratezza superiore all'80% e un'ottima matrice di confusione.

Queste differenze così marcate tra i valori dell'entropia associati ai diversi task però non si sono replicate durante l'esecuzione online. Il fatto che i l'entropia abbia valori così simili sicuramente è una spiegazione del perché il classificatore non riesca a identificare in modo preciso la classe corretta. In particolare dimostra un bias considerevole verso la classe INC: ciò può essere dovuto alla difficoltà che si può avere nel replicare con esattezza gli stessi "movimenti"; inoltre in alcune esecuzioni dei task di movimento durante l'esperimento si è riscontrato come dopo alcuni secondi in cui non si era riusciti a riempire

pienamente la relativa barra il sistema virasse immediatamente sulla classe dell'INC. Quest'ultima casistica può invece essere motivata dalla difficoltà nel mantenere sufficientemente alta l'attenzione per un tempo prolungato. Un altro aspetto che può avvalorare le ipotesi fatte in questa sezione è la grande deviazione standard che si può osservare nella figura 3.11, che è indicativo del fatto che può essere complicato mantenere coerenza tra i vari task di movimento.

4.2 Conclusioni e sviluppi futuri

Si ritiene che la strada intrapresa in questo lavoro di tesi, come già accennato in alcuni punti in precedenza, possa rappresentare un buon metodo per il riconoscimento dell'Intentional Non-Control nel campo delle BCI ma sono necessari miglioramenti al sistema attuale.

Un primo passo potrebbe essere quello di replicare l'esperimento con più soggetti e con più iterazioni: in questo modo i diversi soggetti sarebbero in grado di familiarizzare maggiormente con il sistema, affrontando meglio i diversi task in modo da rendere più facilmente replicabili i diversi pattern. Un'altra considerazione nell'ottica di miglioramento del sistema è quella relativa a un'ulteriore miglioramento del calcolo dell'entropia che risulta particolarmente oneroso quando vengono utilizzate tre o più bande in tempo reale. In questo lavoro il problema è stato risolto grazie al downsampling, permettendo di rimanere in una finestra di 16 Hz (66,5 ms) per la classificazione online, ma per quanto vicina alla curva originale si tratta comunque di un'approssimazione che taglia dei valori dell'entropia (sia in alto che in basso). Riuscire a ottimizzare, o trovare comunque metodi per rendere meno impattante questa operazione sulle performance del sistema potrebbe offrire

un miglioramento delle prestazioni.

Si noti inoltre che durante l'esperimento, delle quattro bande iniziali selezionate per la fase di training, sono state selezionate features solo su una di queste, rendendo quindi il downsampling, in questo caso specifico, non necessario.

Bibliografia

- [1] M. Kyrarini, F. Lygerakis, A. Rajavenkatanarayanan, C. Sevastopoulos, H. R. Nambiappan, K. K. Chaitanya, A. R. Babu, J. Mathew, and F. Makedon, “A survey of robots in healthcare,” Technologies, vol. 9, no. 1, 2021.
- [2] P. Voss, M. E. Thomas, J. M. Cisneros-Franco, and É. de Villers-Sidani, “Dynamic brains and the changing rules of neuroplasticity: implications for learning and recovery,” Frontiers in psychology, vol. 8, p. 1657, 2017.
- [3] J. A. Young and M. Tolentino, “Neuroplasticity and its applications for rehabilitation,” American journal of therapeutics, vol. 18, no. 1, pp. 70–80, 2011.
- [4] M. Alwardat and M. Etoom, “Effectiveness of robot-assisted gait training on freezing of gait in people with parkinson disease: evidence from a literature review,” Journal of exercise rehabilitation, vol. 15, no. 2, p. 187, 2019.
- [5] G. Abbruzzese, R. Marchese, L. Avanzino, and E. Pelosin, “Rehabilitation for parkinson’s disease: Current outlook and future challenges,” Parkinsonism & related disorders, vol. 22, pp. S60–S64, 2016.

-
- [6] M. G. Maggio, M. Russo, M. F. Cuzzola, M. Destro, G. La Rosa, F. Molonia, P. Bramanti, G. Lombardo, R. De Luca, and R. S. Calabrò, “Virtual reality in multiple sclerosis rehabilitation: A review on cognitive and motor outcomes,” Journal of Clinical Neuroscience, vol. 65, pp. 106–111, 2019.
- [7] M. Sebastián-Romagosa, W. Cho, R. Ortner, and C. Guger, “Brain computer interface treatment for gait rehabilitation of stroke patients—preliminary results,” in 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 2948–2951, IEEE, 2021.
- [8] M. Atzori and H. Müller, “Control capabilities of myoelectric robotic prostheses by hand amputees: a scientific research and market overview,” Frontiers in systems neuroscience, vol. 9, p. 162, 2015.
- [9] M. Iosa, G. Morone, A. Cherubini, and S. Paolucci, “The three laws of neurorobotics: a review on what neurorehabilitation robots should do for patients and clinicians,” Journal of medical and biological engineering, vol. 36, no. 1, pp. 1–11, 2016.
- [10] P. Hamet and J. Tremblay, “Artificial intelligence in medicine,” Metabolism, vol. 69, pp. S36–S40, 2017.
- [11] L. Tonin and J. d. R. Millán, “Noninvasive brain–machine interfaces for robotic devices,” Annual Review of Control, Robotics, and Autonomous Systems, vol. 4, pp. 191–214, 2021.
- [12] M. F. Mridha, S. C. Das, M. M. Kabir, A. A. Lima, M. R. Islam, and Y. Watanobe, “Brain-computer interface: Advancement and challenges,” Sensors, vol. 21, no. 17, 2021.

-
- [13] D. Steyrl, R. J. Kobler, G. R. Müller-Putz, et al., “On similarities and differences of invasive and non-invasive electrical brain signals in brain-computer interfacing,” Journal of biomedical science and engineering, vol. 9, no. 08, p. 393, 2016.
- [14] H. H. Jasper, “The ten-twenty electrode system of the international federation,” Electroencephalogr. Clin. Neurophysiol., vol. 10, pp. 370–375, 1958.
- [15] J. S. Kumar and P. Bhuvaneswari, “Analysis of electroencephalography (eeg) signals and its categorization—a study,” Procedia Engineering, vol. 38, pp. 2525–2536, 2012. INTERNATIONAL CONFERENCE ON MODELLING OPTIMIZATION AND COMPUTING.
- [16] R. Abiri, S. Borhani, E. W. Sellers, Y. Jiang, and X. Zhao, “A comprehensive review of eeg-based brain-computer interface paradigms,” Journal of neural engineering, vol. 16, no. 1, p. 011001, 2019.
- [17] S. Aggarwal and N. Chugh, “Review of machine learning techniques for eeg based brain computer interface,” Archives of Computational Methods in Engineering, pp. 1–20, 2022.
- [18] G. Pfurtscheller and C. Neuper, “Motor imagery activates primary sensorimotor area in humans,” Neuroscience letters, vol. 239, no. 2-3, pp. 65–68, 1997.
- [19] S. Sadeghi and A. Maleki, “Recent advances in hybrid brain-computer interface systems: a technological and quantitative review,” Basic and clinical neuroscience, vol. 9, no. 5, p. 373, 2018.

-
- [20] M. Wang, I. Daly, B. Z. Allison, J. Jin, Y. Zhang, L. Chen, and X. Wang, "A new hybrid bci paradigm based on p300 and ssvep," Journal of neuroscience methods, vol. 244, pp. 16–25, 2015.
- [21] S. Abenna, M. Nahid, H. Bouyghf, and B. Ouacha, "Eeg-based bci: A novel improvement for eeg signals classification based on real-time preprocessing," Computers In Biology And Medicine, vol. 148, p. 105931, 2022.
- [22] B. Graimann, B. Allison, and G. Pfurtscheller, "Brain–computer interfaces: A gentle introduction," Brain-computer interfaces: Revolutionizing human-computer interaction, pp. 1–27, 2010.
- [23] J. J. Vidal, "Toward direct brain-computer communication," Annual review of Biophysics and Bioengineering, vol. 2, no. 1, pp. 157–180, 1973.
- [24] A. Kübler, "The history of bci: From a vision for the future to real support for personhood in people with locked-in syndrome," Neuroethics, vol. 13, no. 2, pp. 163–180, 2020.
- [25] H. Wang, T. Li, A. Bezerianos, H. Huang, Y. He, and P. Chen, "The control of a virtual automatic car based on multiple patterns of motor imagery bci," Medical & biological engineering & computing, vol. 57, pp. 299–309, 2019.
- [26] N. Jiang, L. Gizzi, N. Mrachacz-Kersting, K. Dremstrup, and D. Farina, "A brain–computer interface for single-trial detection of gait initiation from movement related cortical potentials," Clinical Neurophysiology, vol. 126, no. 1, pp. 154–159, 2015.

-
- [27] E. Dagdevir and M. Tokmakci, "Optimization of preprocessing stage in eeg based bci systems in terms of accuracy and timing cost," Biomedical Signal Processing and Control, vol. 67, p. 102548, 2021.
- [28] K. Renuga Devi and H. Hannah Inbarani, "Neighborhood based decision theoretic rough set under dynamic granulation for bci motor imagery classification," Journal on Multimodal User Interfaces, pp. 1–21, 2021.
- [29] P. Arpaia, A. Esposito, A. Natalizio, and M. Parvis, "How to successfully classify eeg in motor imagery bci: a metrological analysis of the state of the art," Journal of Neural Engineering, 2022.
- [30] L. Tonin, A. Cimolato, and E. Menegatti, "Do not move! entropy driven detection of intentional non-control during online smr-bci operations," in Converging Clinical and Engineering Research on Neurorehabilitation II: Proceedings of the 3rd International Conference on NeuroRehabilitation (ICNR2016), October 18-21, 2016, Segovia, Spain, pp. 989–993, Springer, 2017.
- [31] J. Ramadoss, N. M. Dawi, K. Rajagopal, and H. Namazi, "Complexity and information-based analysis of the electroencephalogram (eeg) signals in standing, walking, and walking with a brain-computer interface," Fractals, vol. 30, no. 01, p. 2250041, 2022.
- [32] L. Tonin, G. Beraldo, S. Tortora, and E. Menegatti, "Ros-neuro: A common middleware for neurorobotics,"
- [33] L. Tonin, G. Beraldo, S. Tortora, L. Tagliapietra, J. d. R. Millán, and E. Menegatti, "Ros-neuro: A common middleware for bmi and robotics. the acquisition and recorder packages," in 2019 IEEE International

- Conference on Systems, Man and Cybernetics (SMC), pp. 2767–2772, IEEE, 2019.
- [34] S. Tortora, G. Beraldo, L. Tonin, and E. Menegatti, “Entropy-based motion intention identification for brain-computer interface,” in 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC), pp. 2791–2798, IEEE, 2019.
- [35] S. Tortora, F. Artoni, L. Tonin, C. Chisari, E. Menegatti, and S. Micera, “Discrimination of walking and standing from entropy of eeg signals and common spatial patterns,” in 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 2008–2013, IEEE, 2020.
- [36] Q. Gu, Z. Li, and J. Han, “Generalized fisher score for feature selection,” arXiv preprint arXiv:1202.3725, 2012.
- [37] V. W. Berger and Y. Zhou, “Kolmogorov–smirnov test: Overview,” Wiley statsref: Statistics reference online, 2014.