



UNIVERSITÀ
POLITECNICA
DELLE MARCHE

FACOLTÀ DI INGEGNERIA

CORSO DI LAUREA IN INGEGNERIA BIOMEDICA

**DESIGN ED ESECUZIONE DI JURY TEST ACUSTICI
ACCOPPIATI CON ANALISI DELLA RISPOSTA FISIOLOGICA
EEG DEI GIURATI**

*Design and execution of acoustic Jury tests coupled with analysis of the
physiological EEG response of the jurors*

Relatore:

Prof.ssa Milena Martarelli

Correlatore:

Prof. Paolo Castellini

Tesi di laurea di:

Laura Nucciarone

A.A. 2023/2024

Abstract

Capitolo 1

Introduzione 1

1.1 EEG 3

1.1.1 Ritmi cerebrali..... 5

Capitolo 2

Materiali e metodi..... 8

2.1 Protocollo sperimentale 8

2.1.1 Partecipanti 8

2.1.2 Muse S..... 9

2.1.3 EmotiBit 12

2.1.4 Sequenza test 14

Capitolo 3

Elaborazione dei dati..... 23

3.1 Pre-processing dati acquisiti 23

3.2 Pre-processing dati EEG 28

3.2.1 Filtraggio dati EEG..... 28

3.3 Analisi PSD	34
-----------------------	----

Capitolo 4

Conclusioni	39
--------------------------	-----------

Appendice A	I
--------------------------	----------

Appendice B	XIII
--------------------------	-------------

Bibliografia

Abstract

Questo studio si concentra sull'analisi della risposta cerebrale agli stimoli acustici attraverso un approccio sperimentale basato su Jury test acustici e registrazioni EEG. L'obiettivo è quello di valutare se esistano correlazioni tra l'attività cerebrale e la percezione soggettiva dei suoni, cercando di integrare dati neurofisiologici e preferenze espresse dai partecipanti.

L'esperimento ha coinvolto 109 soggetti, i quali hanno partecipato ad un test suddiviso in due fasi. Nella prima, i partecipanti dovevano scegliere quale suono percepivano più piacevole tra i due proposti; nella seconda, invece, erano esposti passivamente a sequenze di suoni piacevoli e fastidiosi. Durante il test, l'attività cerebrale è stata monitorata con il dispositivo Muse S, mentre altri parametri fisiologici, come attività elettrodermica e frequenza cardiaca, sono stati acquisiti con EmotiBit.

Per analizzare i dati acquisiti è stato necessario segmentare i segnali in base ai timestamp registrati durante il test. Sono state poi assegnate etichette che identificano le varie fasi sperimentali e le scelte compiute dai partecipanti. Il segnale EEG è stato sottoposto a pre-processing, applicando filtri per ridurre il rumore e migliorare la qualità dell'analisi.

La densità spettrale di potenza (PSD) è stata stimata con il metodo di Welch per analizzare la distribuzione dell'energia nelle diverse bande di frequenza. Il confronto tra le PSD dei singoli soggetti e la media complessiva ha evidenziato una certa variabilità individuale, ma ha confermato una tendenza generale: l'audio fastidioso mostra un incremento della potenza nelle frequenze medio-basse, soprattutto nei canali frontali, mentre la condizione di silenzio presenta livelli inferiori di PSD. Questi risultati suggeriscono un effetto degli

stimoli acustici sull'attività cerebrale, evidenziando differenze nella risposta neurale tra le diverse condizioni sonore.

Capitolo 1

Introduzione

Negli ultimi anni, nel campo dell'ingegneria acustica, l'interesse verso la qualità del suono e la percezione acustica ha acquisito una crescente importanza, soprattutto andando ad integrare tale analisi con dispositivi indossabili per il rilevamento di segnali elettroencefalografici e altri parametri fisiologici.

L'approccio utilizzato in questa tesi mira a fornire un'analisi più oggettiva, affidabile e accurata della percezione uditiva umana.

La qualità del suono rappresenta un aspetto critico in numerosi ambiti, dall'audio consumer all'audiologia clinica. È generalmente valutata sia attraverso parametri oggettivi come distorsione, rapporto segnale-rumore (SNR) e risposta in frequenza, ma anche mediante l'esperienza soggettiva degli utenti. Quest'ultima, infatti, permette di valutare l'influenza non solo dalle caratteristiche fisiche del suono, ma anche dei fattori cognitivi e neurofisiologici, in modo da poter valutare e comprendere quali caratteristiche delle onde cerebrali sono associate a suoni piacevoli o, al contrario, a suoni fastidiosi. [1]

L'integrazione di sensori EEG in dispositivi indossabili ha aperto nuove prospettive per lo studio della percezione acustica. A differenza dei tradizionali sistemi EEG, i wearable EEG sono progettati per essere leggeri, comodi e facili da utilizzare, rendendoli adatti ad un monitoraggio continuo.

Questi dispositivi consentono di raccogliere dati neurofisiologici in tempo reale durante l'esposizione a diversi tipi di stimoli acustici. Ad esempio, è possibile analizzare come varia l'attività cerebrale in risposta a suoni con differenti caratteristiche timbriche o a diverse condizioni di rumore ambientale. Tali informazioni possono essere utilizzate per migliorare la progettazione di dispositivi audio e per personalizzare l'esperienza sonora.

La percezione acustica soggettiva viene convenzionalmente misurata attraverso metriche di qualità del suono determinate mediante un processo di correlazione con i risultati dei test effettuati su un gruppo di ascoltatori (Jury test). L'utilizzo di misurazioni elettroencefalografiche (EEG) durante i Jury test, per registrare l'attività cerebrale in risposta agli stimoli acustici presentati ai giurati, può permettere di stimare la percezione dei giurati direttamente dalla loro risposta fisiologica.

1.1 EEG

L'elettroencefalogramma, comunemente detto EEG, consiste nella registrazione dell'attività elettrica del cervello, attraverso l'applicazione di un certo numero di elettrodi. Il segnale ottenuto è la somma dell'attività elettrica di ampie popolazioni di neuroni, in quanto l'attivazione di un singolo neurone non è rilevabile a causa della sua bassa entità. Il tracciato prodotto dall'EEG è una registrazione grafica dell'attività del cervello, la quale non è coordinata con precisione temporale, né periodica: è possibile avere più funzioni neurologiche attive contemporaneamente che generano un segnale elettrico non ripetibile neppure quando quella funzione viene ripetuta con la stessa intensità; nasce quindi la difficoltà nell'analisi di tale tracciato.

L'attività elettrica può essere misurata attraverso tre tipi di elettrodi:

- Elettrodi posizionati sullo scalpo (EEG standard di superficie)
- Elettrodi ad ago, i quali vengono inseriti in profondità nel tessuto cerebrale (Stereoelettroencefalografia "SEEG")
- Elettrodi posti sulla superficie della corteccia cerebrale, la quale deve essere esposta (Elettrocorticografia "ECoG")

Gli elettrodi utilizzati per la rilevazione superficiale sono in Ag/AgCl e sono ricoperti con un gel elettro-conduttore il quale favorisce la conduzione del segnale tra l'elettrodo e la cute. Tali elettrodi vengono posizionati su una cuffia elastica precedentemente posizionata sulla testa del paziente; viene utilizzata una pasta abrasiva che permette di pulire la cute, rimuovendo cellule morte e sebo che potrebbero compromettere la trasmissione del segnale tra l'elettrodo e la cute.

L'EEG può essere registrato utilizzando un monitoraggio monopolare o bipolare: nel caso monopolare viene registrata la differenza di potenziale tra un elettrodo attivo sullo scalpo e uno di riferimento solitamente posizionato in uno dei due lobi dell'orecchio; nel caso bipolare viene invece rilevata la differenza di tensione tra due elettrodi attivi posti sullo scalpo.

Il sistema di derivazioni internazionalmente noto è il sistema 10-20 (Figura 5), il quale si basa sulla disposizione di 20 elettrodi sullo scalpo (10 nella parte sinistra e 10 nella parte destra). Tutti gli elettrodi hanno una codifica alfanumerica specifica: regione centrale **C**, regione frontale **F**, regione temporale **T**, regione parietale **P**, regione occipitale **O** e regione fronto-polare **Fp**; inoltre, i numeri pari rappresentano gli elettrodi connessi all'emisfero destro, mentre i numeri dispari a quello sinistro.

1.1.1 Ritmi cerebrali

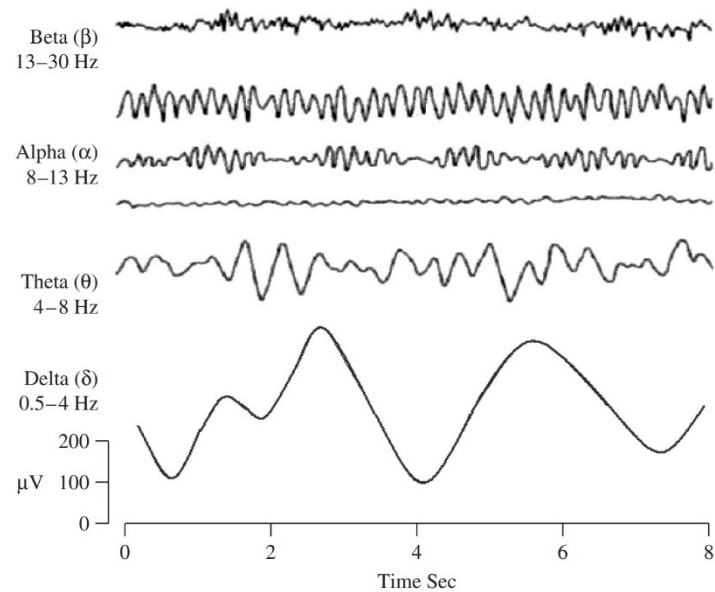


Figura 1: segnali EEG nelle varie bande di frequenza [2]

L'estensione della banda del segnale EEG è circa tra 0.5-100 Hz, ma la maggior parte del contenuto informativo si trova entro i 40 Hz. Il segnale EEG, come precedentemente detto, è caratterizzato da un segnale di fondo privo di ritmicità, ma è possibile distinguere alcune oscillazioni ritmiche sulla base della frequenza di oscillazione, come si vede nella Figura 1. Le onde EEG sono infatti denominate in base al loro intervallo di frequenza utilizzando lettere greche:

- Onde delta, con intervallo di frequenza compreso tra 0.5-4 Hz. Sono associate normalmente durante lo stato di sonnolenza e nelle prime fasi del sonno, mentre se presenti durante lo stato di veglia possono essere indicative di una disfunzione cerebrale. È molto facile confondere gli artefatti dovuti a muscoli di naso e mascella con le onde delta, questo perché tali muscoli sono vicini alla superficie della pelle, mentre il segnale di interesse viene ampiamente attenuato passando

attraverso la scatola cranica. Tramite semplici analisi sul segnale è possibile riconoscere se esso sia causato da movimenti o meno. [2]

- Onde theta, con banda di frequenza compresa tra 4-8 Hz. Sono particolarmente evidenti in certe fasi di sonno; esse sono state associate ad accesso all'inconscio, all'ispirazione creativa e profonda meditazione. Le onde theta sono state anche associate a meccanismi di memorizzazione. L'onda theta gioca anche un importante ruolo nell'infanzia ed adolescenza; cambiamenti di tale onda sono anche esaminati in studi emozionali e di maturazione. [2]
- Onde alpha, con banda di frequenza compresa tra 8-12 Hz. Hanno ampiezza maggiore nelle aree occipitali e un'ampiezza normalmente inferiore a 50 mV. Tali onde si manifestano con una forma piuttosto sinusoidale e costituiscono l'attività predominante negli adulti durante lo stato di veglia rilassato con occhi chiusi.
- Onde beta hanno un'ampiezza minore e una frequenza compresa tra 12-30 Hz. Sono più evidenti nelle aree frontali e a livello di altre regioni durante attività mentali intense come parlare e leggere. Pertanto, sono associate all'attivazione fisiologica (eccitazione), all'attenzione, al pensiero analitico, alla risoluzione di compiti complessi e ad altri processi intellettuali.
- Onde gamma, con banda di frequenza compresa tra 30-50 Hz. Sono onde molto veloci e, a causa della loro piccola ampiezza, non sono facili da registrare. Identificate più recentemente rispetto alle altre onde, sono finora meno note ed emergono principalmente durante momenti di profonda concentrazione e di massima prestazione, sia fisica che mentale, nonché durante esperienze mistiche e trascendentali.

In conclusione, gli studi hanno identificato le onde cerebrali come una risorsa sempre più preziosa per comprendere i meccanismi cerebrali, sia nei cervelli sani che in quelli affetti da patologie. Pertanto, il loro studio approfondito può rappresentare una nuova frontiera negli approcci di studio e intervento neuroscientifici e neuroriabilitativi.

Capitolo 2

Materiali e metodi

2.1 Protocollo sperimentale

2.1.1 Partecipanti

Studenti, docenti e personale accademici (totale soggetti: 109, 47 donne e 62 uomini) dell'Università Politecnica delle Marche hanno partecipato all'esperimento in maniera volontaria.

Come mostrato in Figura 2 ai partecipanti è stato richiesto di compilare un calendario condiviso su Excel, indicando il proprio nome, cognome ed e-mail nello slot corrispondente all'orario di disponibilità.

Dopo aver firmato l'informativa scritta esaustiva di dettagli, si è proceduto con il test.

Lo studio e le procedure sono stati approvati dal comitato etico.

Giorno	Ora	Nome	Cognome	Email	Username	Note
Mercoledì 20 novembre	9.00-9.30				S52	
	9.30-10.00				S53	
	10.00-10.30				S54	
	10.30-11.00				S55	Emotibit strano, si connetteva e disconnetteva
	11.00-11.30				S56	No emotibit
	11.30-12.00				S57	
	12.00-12.30					
	12.30-13.00					
	14.00-14.30				Febbre	
	14.30-15.00				S58	
	15.00-15.30					
	15.30-16.00				S59	
	16.00-16.30				S60	
	16.30-17.00				S61	
	17.00-17.30				S62	

Figura 2: organizzazione calendario Excel

2.1.2 Muse S

In questo studio l'acquisizione del segnale EEG è stata effettuata tramite il dispositivo Muse S, mostrato in Figura 3. Il Muse S è una fascia indossabile progettata per rilevare l'attività cerebrale tramite elettroencefalografia (EEG), disponibile per uso commerciale per scopi come: meditazione, rilassamento e il monitoraggio del sonno; si tratta di un dispositivo compatto dal design ultra-portatile, il quale semplifica notevolmente l'accesso e l'utilizzo dei dati delle onde cerebrali. Inoltre, i sistemi piccoli e wireless, con elettrodi a secco che non richiedono una preparazione della cute, possono aumentare la tolleranza e l'accettabilità degli utenti. [3]

Il Muse S è dotato di quattro canali EEG e due canali ausiliari amplificati, con una frequenza di campionamento di 256 Hz e una risoluzione di 12 bit per campione (è possibile rilevare variazioni molto sottili nel segnale EEG, con accuratezza elevata).



Figura 3: Muse S [4]

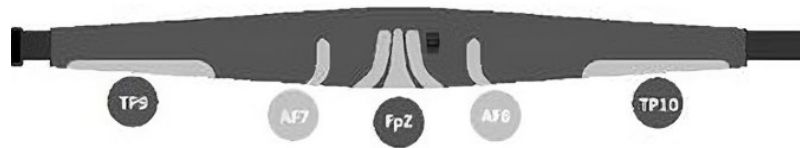


Figura 4: posizione degli elettrodi per l'acquisizione del segnale EEG sulla banda Muse S [5]

La Figura 4 e la Figura 5 mostrano la posizione dei canali prelevati dagli elettrodi del Muse secondo lo standard 10-20 e sono: due temporoparietali (TP9 e TP10) e due frontali (AF7 e AF8), con l'elettrodo di riferimento posizionato in Fpz.

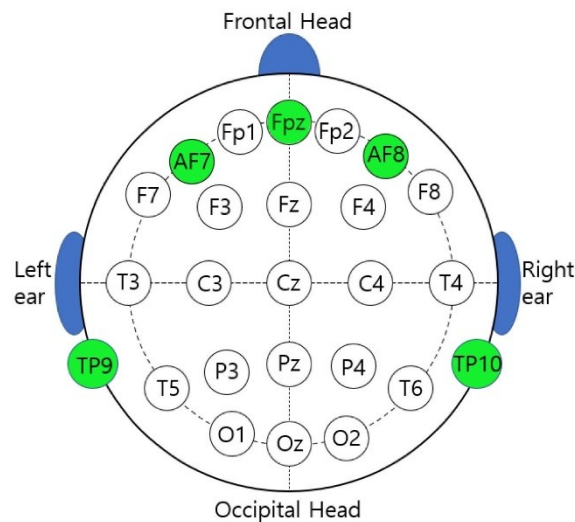


Figura 5: configurazione sistema 10-20 degli elettrodi con vista dall'alto, in verde sono evidenziati i canali corrispondenti a quelli del Muse S [6]

Sebbene il numero limitato di elettrodi possa inizialmente far dubitare sulla qualità del segnale registrato, diversi studi confermano che, in presenza di un campione numeroso di soggetti, l'analisi dei dati risulta comunque completa.

Sono stati effettuati vari studi in cui sono state messe a confronto tipologie di acquisizione classiche con quelle low cost, in particolare con il Muse. Il risultato emerso è quello che, nei sistemi low cost le anomalie nel segnale si riscontrano in maniera più frequente negli

elettrodi frontali, in quanto più sensibili ad artefatti di blinking e di movimento muscolare nella regione frontale. D'altra parte, nel calcolo della potenza spettrale, entrambe le tipologie di strumenti risultano approssimativamente uguali, eccetto nel caso della potenza a banda larga che risulta essere maggiore nel Muse; questo, potrebbe essere giustificato dal fatto che vengono utilizzati elettrodi a secco. [7]

Inoltre, è emerso come alcune caratteristiche anatomiche, come la forma o le dimensioni della testa, o particolari stili di capelli, possano ostacolare l'acquisizione dei dati con il Muse [8]; ai soggetti di sesso femminile, in particolare è stato richiesto di legare i capelli in maniera tale da ridurre il volume dei capelli al di sotto della fascia Muse.

Un altro aspetto da notare riguardo il dispositivo Muse è il jitter temporale rilevato nei dati acquisiti, si attesta infatti un ritardo da 18 a 20 ms e un jitter di 20 ms durante le connessioni Bluetooth. [9] Tuttavia, questo non rappresenta un punto di debolezza in questo studio poiché vengono effettuate valutazioni quantitative.

I dati sono stati registrati in forma grezza e salvati in formato .csv, per un'elaborazione offline e successivamente sono stati applicati algoritmi di analisi dati.

2.1.3 EmotiBit

EmotiBit è un sensore indossabile per catturare dati emozionali, fisiologici e di movimento di alta qualità, i quali vengono trasmessi in modalità wireless e archiviati localmente. Si tratta dunque di un dispositivo multiparametrico posizionato, in tale studio, sul polso del soggetto (Figura 6).

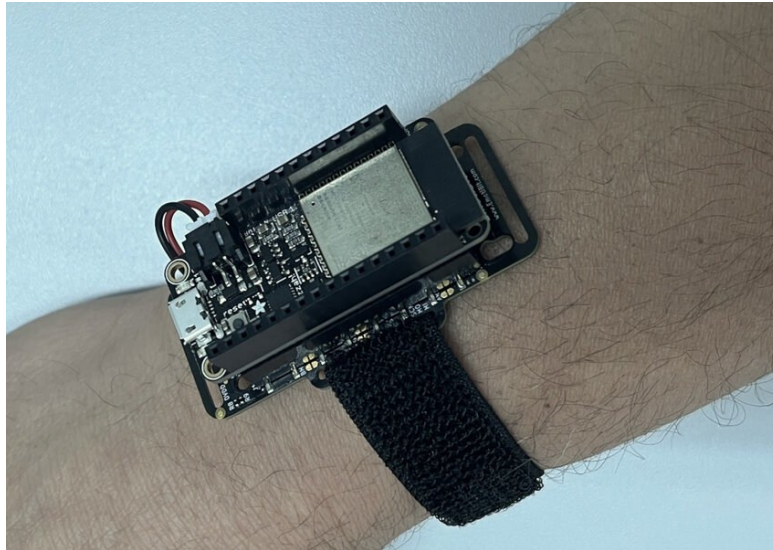


Figura 6: EmotiBit posizionato sul polso di un giurato

I dati raccolti da Emotibit riguardano l'attività elettrodermica (EDA), la fotopletismografia multi-lunghezza d'onda (PPG), un sensore di temperatura e un sensore di unità di misura inerziale (IMU) a 9 assi.

I dati acquisiti dall'Emotibit sono così suddivisi:

- file *EDA_data* (frequenza di campionamento 25 Hz) comprende i dati *eda* (Electrodermal Activity), *edl* (Electrodermal Level)

- file *Motion_data* (frequenza di campionamento 25 Hz) comprende i dati $\{acx, acy, acz\}$ dell'accelerometro, i dati $\{gyx, gyy, gyz\}$ del giroscopio e i dati $\{mgx, mgy, mgz\}$ del magnetoscopio
- file *PPG_data* (frequenza di campionamento 25 Hz) comprende i dati pgi (PPG Infrared), pgr (PPG Red) e pgg (PPG Green); raccolti quindi con lunghezze d'onda differenti
- file *Temperature_data* (frequenza di campionamento 7.5 Hz) comprende i dati thr.

Il sensore EDA funziona rilevando variazioni nell'attività elettrica (ionica) attribuibili a fluttuazioni nella funzione delle ghiandole sudoripare; gli elettrodi devono avere una certa sensibilità a tali cambiamenti ed efficacia nel trasmettere i dati all'apparato di registrazione. Gli elettrodi EDA di Emotibit presentano un'interfaccia di contatto Ag/AgCl (cloruro di argento) con la pelle.

L'attività elettrodermica (EDA), nota anche come risposta galvanica cutanea (GSR), denota l'alterazione della conduttanza elettrica cutanea innescata dalla secrezione di sudore, spesso in minuscole quantità. Questi dati fisiologici vengono raccolti tramite l'applicazione di una tensione bassa, impercettibile e costante alla pelle, seguita dalla valutazione delle variazioni nella conduttanza cutanea. Il segnale EDA funge da indicatore dell'intensità dell'eccitazione emozionale, che cambia dinamicamente in risposta agli stimoli ambientali, siano essi allarmanti, gioiosi o emotivamente salienti. Le ricerche hanno chiarito come sia gli stimoli positivi che quelli negativi suscitino un'eccitazione accresciuta e una conduttanza cutanea aumentata, riflettendo varie dimensioni della risposta emotiva. Di conseguenza, il segnale EDA non delinea in modo categorico i tipi emotivi, ma piuttosto ne indica l'intensità. [10]

2.1.4 Sequenza test

Lo svolgimento del test sui partecipanti è stato eseguito seguendo un calendario di prenotazione prestabilito, come mostrato in Figura 2.

La durata complessiva del test rispettava un tempo inferiore ai trenta minuti, inclusi della preparazione e della conclusione del test.

All'arrivo dei soggetti in laboratorio veniva illustrato loro il modo in cui si sarebbe sviluppato il test; in questa fase preliminare veniva inoltre richiesto di firmare il consenso informato, necessario per la partecipazione al test e per la registrazione del test.

Successivamente, i soggetti venivano invitati ad accomodarsi nella postazione stabilita, seduti davanti ad un computer destinato alla somministrazione del test. Per ogni partecipante è stato necessario creare un profilo anagrafico sulla piattaforma *Wepop* (Figura 7) in modo tale da garantire il salvataggio dei dati acquisiti in modo univoco tramite uno username.

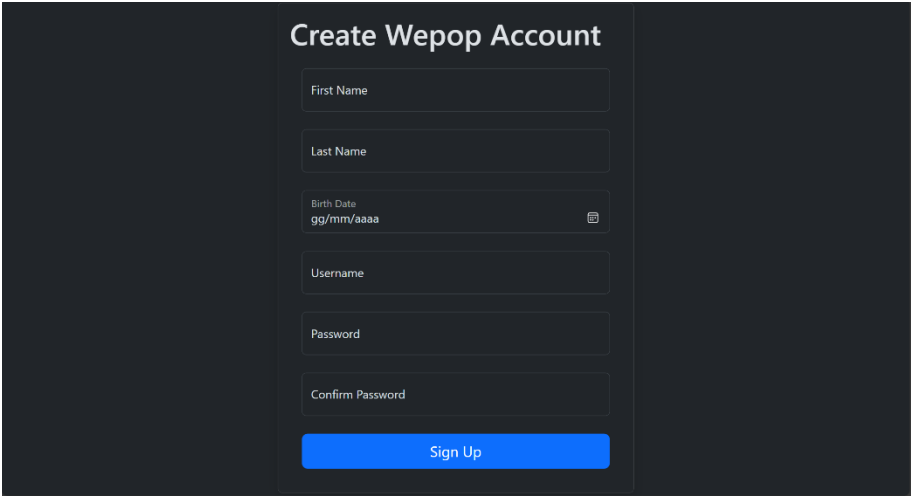
The image shows a dark-themed web interface for creating a Wepop account. The title 'Create Wepop Account' is at the top. Below it are several input fields: 'First Name', 'Last Name', 'Birth Date' (with a date picker icon and placeholder 'gg/mm/aaaa'), 'Username', 'Password', and 'Confirm Password'. At the bottom is a blue 'Sign Up' button.

Figura 7: schermata per inserimento dati anagrafici del soggetto

Mentre veniva completata la registrazione del profilo sulla piattaforma *Wepop*, ai soggetti venivano posizionati i sensori necessari per la raccolta dei dati fisiologici e delle cuffie on-ear connesse con apposito jack al computer (Figura 9). In particolare, per ridurre il rischio di un errato posizionamento degli elettrodi del dispositivo Muse S, si è fatto uso del supporto dell'app Muse, la quale, tramite connessione Bluetooth, permette di verificare il corretto posizionamento degli elettrodi della fascia: l'app permette di monitorare l'impedenza degli elettrodi accertandosi che l'icona associata a ciascuno dei quattro elettrodi risulti completamente colorata in verde, segnalando così un posizionamento ottimale, come si vede in Figura 8.

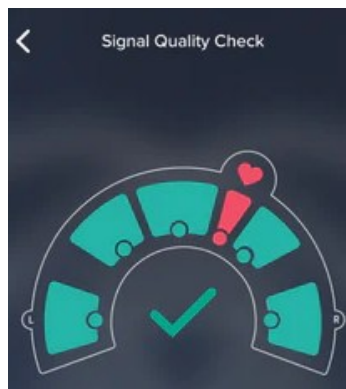
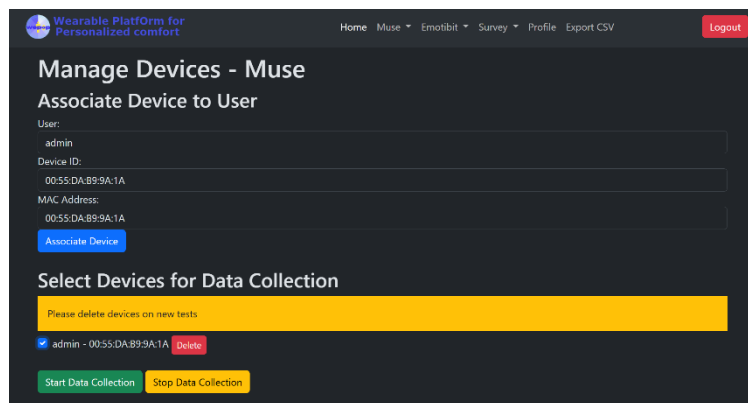


Figura 8: icona per la verifica dell'impedenza degli elettrodi tramite app Muse



Figura 9: soggetto equipaggiato con sensori e cuffie

Una volta posizionati i dispositivi per l'acquisizione dei segnali fisiologici, prima dell'inizio del test, devono essere associati al profilo del soggetto (Figura 10 e Figura 11) e deve essere verificata anche l'effettiva acquisizione dei dati per garantire l'operatività dei suddetti (Figura 12 e Figura 13).



Wearable Platform for Personalized comfort

Home Muse Emotibit Survey Profile Export CSV Logout

Manage Devices - Muse

Associate Device to User

User: admin

Device ID: 00:55:DA:89:9A:1A

MAC Address: 00:55:DA:89:9A:1A

Associate Device

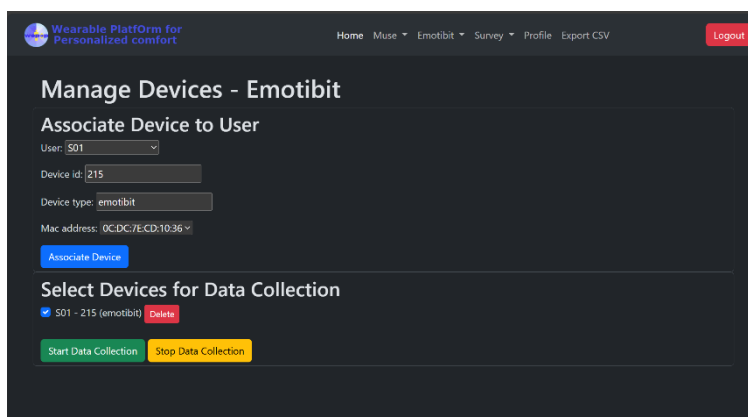
Select Devices for Data Collection

Please delete devices on new tests

☒ admin - 00:55:DA:89:9A:1A Delete

Start Data Collection Stop Data Collection

Figura 10: associazione Muse sulla piattaforma Wepop



Wearable Platform for Personalized comfort

Home Muse Emotibit Survey Profile Export CSV Logout

Manage Devices - Emotibit

Associate Device to User

User: S01

Device id: 215

Device type: emotibit

Mac address: 0C:DC:7E:CD:10:36

Associate Device

Select Devices for Data Collection

Please delete devices on new tests

☒ S01 - 215 (emotibit) Delete

Start Data Collection Stop Data Collection

Figura 11: associazione EmotiBit sulla piattaforma Wepop

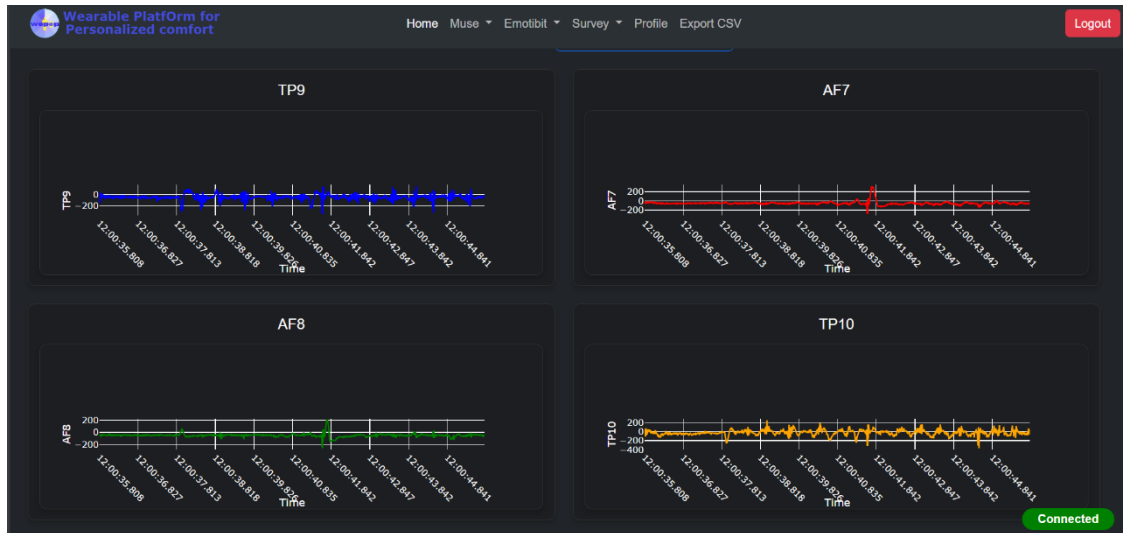


Figura 12: controllo della corretta associazione del dispositivo Muse

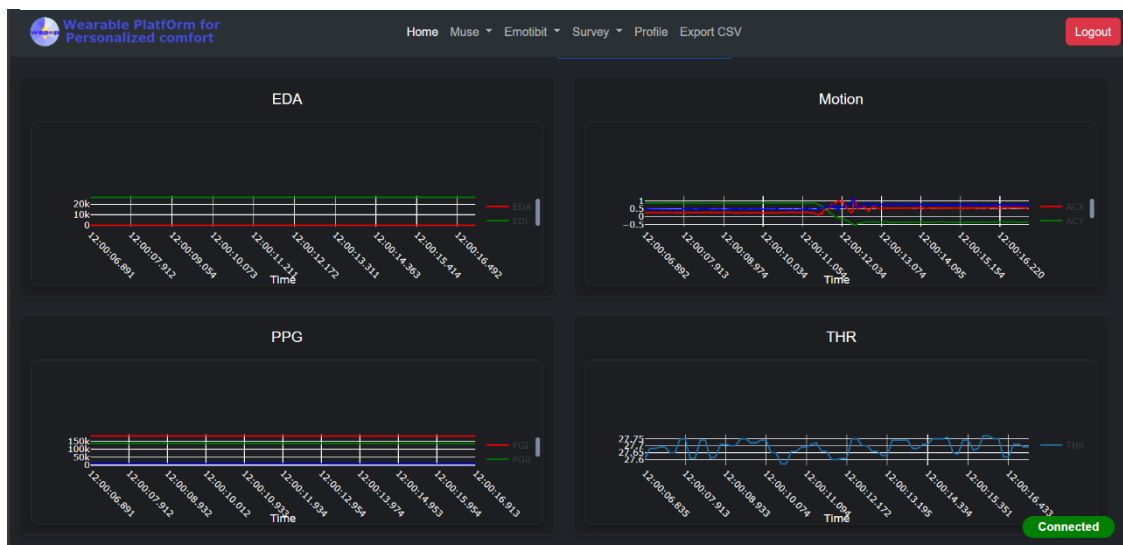


Figura 13: controllo della corretta associazione del dispositivo EmotiBit

Al fine di evitare movimenti che possano interferire con la raccolta dei dati, veniva richiesto di mantenere una posizione il più ferma possibile. Inoltre, veniva raccomandato al soggetto di liberare la mente da possibili distrazioni o pensieri esterni, mantenendo l'attenzione esclusivamente sull'attività proposta per garantire risultati affidabili e accurati.

Un'ulteriore richiesta effettuata al soggetto era quella di impostare il volume del computer su un valore pari a 30, in modo da garantire uniformità nel test tra tutti i soggetti; tuttavia se l'audio risultava eccessivamente elevato, il soggetto poteva diminuire l'audio con l'accortezza di riferirlo a test ultimato.

Dopo aver configurato l'intero assetto per lo svolgimento del test, il giurato veniva lasciato solo all'interno della stanza in modo da eliminare ogni forma di disturbo esterno. In questo ambiente controllato il giurato può iniziare il test in condizioni ottimali di concentrazione e comfort.

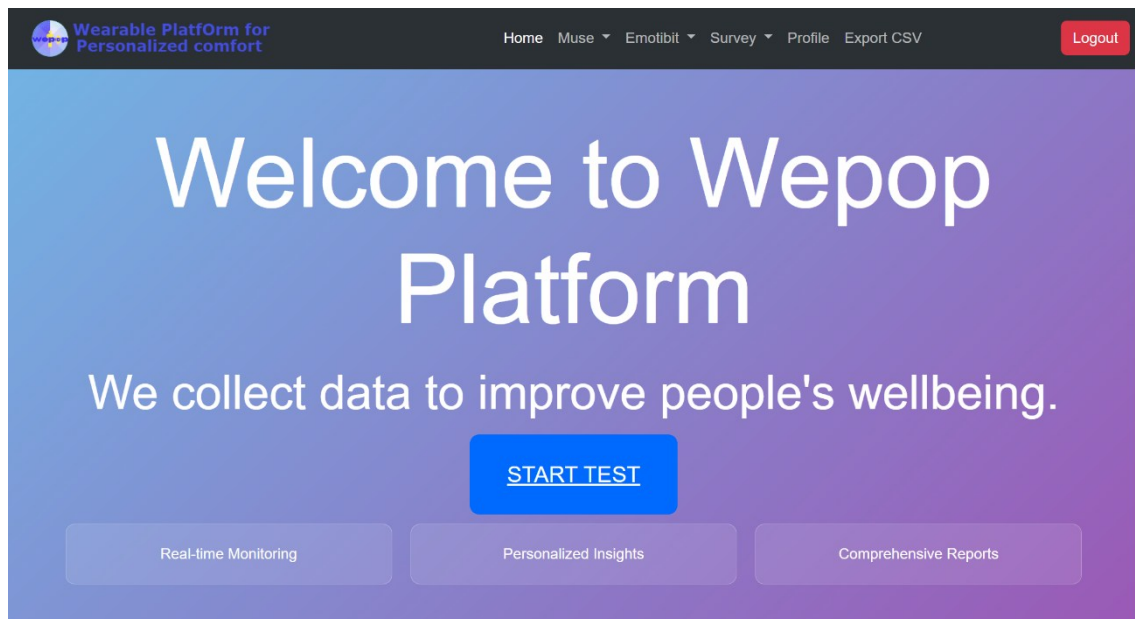


Figura 14: schermata per l'avvio del test

Nel momento in cui il soggetto cliccava il pulsante "START TEST" (Figura 14) il test si considerava avviato. Sullo schermo compariva immediatamente un led verde che funge da feedback visivo per il giurato indicando che il test è attivo; tale led permane fino al termine del test. Contestualmente viene mostrata una schermata nera durante la quale non

viene riprodotto alcun suono, dalla durata di 60 secondi, permettendo al soggetto di prepararsi e concentrarsi completamente sull'esperienza.

Dopo i primi 60 secondi, al soggetto è richiesto di cliccare il tasto “INIZIA IL TEST” (Figura 15) in questo momento nel file S(*)_data .csv viene salvato il primo timestamp nel formato “dd-MM-yyyy HH:mm:ss.SSS”.

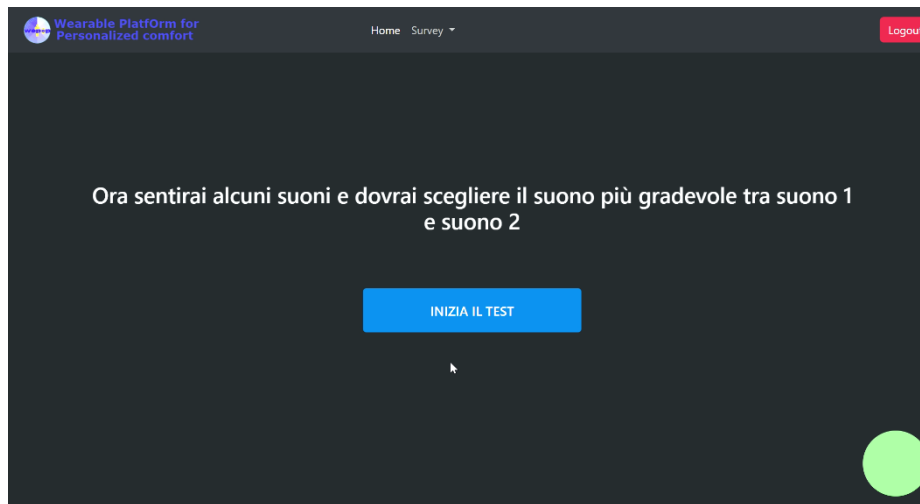


Figura 15: schermata dopo la quale inizia la prima fase del test

Il test consiste di due parti. Nella prima parte venivano presentate 12 coppie di suoni: al termine dell'ascolto della coppia, il soggetto doveva effettuare attivamente una scelta indicando il suono di sua preferenza. La seconda parte, invece, consisteva in un semplice ascolto passivo, durante il quale venivano esclusivamente registrati i segnali rilevati dai sensori.

Prima parte: test di confronto di suoni automobilistici (registrati su auto elettriche o con motore a combustione). La sequenza, ad esempio per la coppia di suoni A1A2, prevede l'ascolto dell'audio A1 con durata di 30 secondi, un silenzio di 2 secondi e

successivamente l'ascolto dell'audio A2 con durata di 30 secondi. A seconda dell'audio ascoltato, il bordo dell'icona corrispondente veniva evidenziato, come mostrato nella Figura 16 e nella Figura 17. Al termine dell'ascolto dell'audio A2, sullo schermo del computer appariva una schermata (Figura 18) in cui veniva chiesto al giurato di indicare quale suono preferiva tra i due appena ascoltati (Jury test).



Figura 16: il bordo relativo al Suono 1 appare evidenziato indicando quindi l'esecuzione del suddetto suono

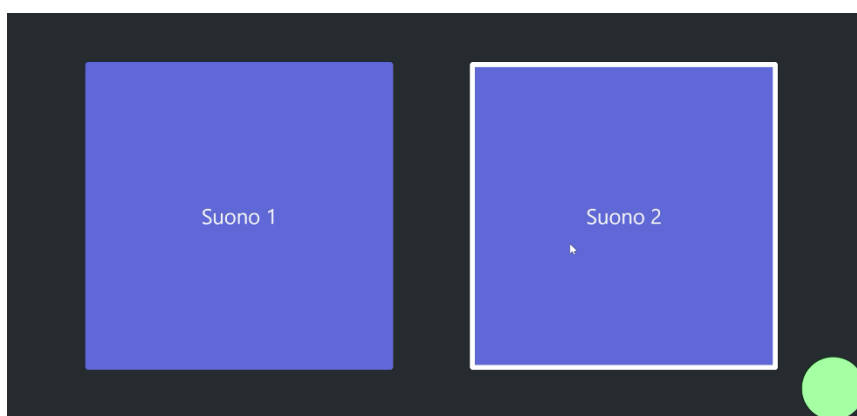


Figura 17: il bordo relativo al Suono 2 appare evidenziato indicando quindi l'esecuzione del suddetto suono



Figura 18: i tasti relativi ai due suoni diventano cliccabili e il giurato può esprimere la propria preferenza

Non appena viene eseguita la scelta tra la coppia di suoni viene salvato il relativo timestamp e viene immediatamente avviata la riproduzione di una nuova coppia da confrontare. Questo procedimento viene ripetuto per un totale di 12 volte, rispettando gli intervalli temporali stabiliti descritti precedentemente.

Seconda parte: sequenza di ascolto. Dopo aver cliccato il pulsante “OK” presente sullo schermo (Figura 19) viene salvato il timestamp corrispondente e inizia una sequenza prestabilita di suoni: 60 secondi di silenzio, seguiti da 60 secondi di suono piacevole, immediatamente seguiti da 60 secondi di suono fastidioso. Successivamente 30 secondi di silenzio separano la riproduzione inversa della sequenza, in particolare: 60 secondi di suono fastidioso, seguiti da 60 secondi di suono piacevole e in conclusione 60 secondi di silenzio.

Al completamento di questa seconda fase il test è concluso e compare una schermata che informa il partecipante della conclusione del test; automaticamente viene salvato l’ultimo timestamp del test e il led verde si spegne.

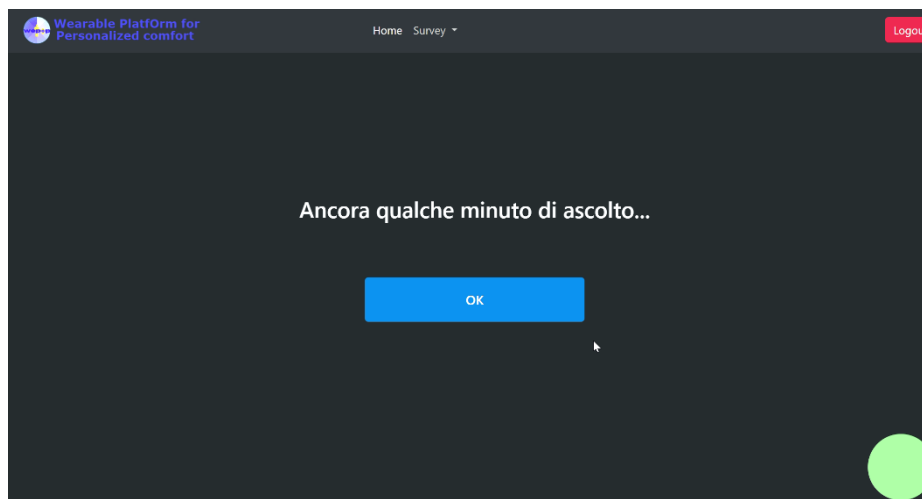


Figura 19: schermata per l'avvio della seconda parte del test

Al termine del test di ciascun soggetto, dopo la rimozione dei vari sensori, nel file Excel di prenotazione (Figura 2), nella riga corrispondente a ogni partecipante, sono state riportate eventuali annotazioni relative a problematiche riscontrate durante il test. Tra queste, ad esempio: batteria dell'Emotibit o Muse S scarica, volume ridotto o rumore significativo su uno specifico canale del segnale EEG.

Tutti i timestamp generati automaticamente durante il test di un determinato soggetto vengono salvati in un file Excel, che registra anche le scelte effettuate nella prima fase del test (Figura 20).

1	Nome	Cognome	Username	Data di nascita	Audio	Scelta	Timestamp
2	*****	*****	S10	*****	start		06-11-2024 10:00:22.711715
3	*****	*****	S10	*****	A1A2	2	06-11-2024 10:01:28.571029
4	*****	*****	S10	*****	A1A3	2	06-11-2024 10:02:35.544486
5	*****	*****	S10	*****	A1A4	2	06-11-2024 10:03:42.282760
6	*****	*****	S10	*****	A2A1	1	06-11-2024 10:04:47.850678
7	*****	*****	S10	*****	A2A3	2	06-11-2024 10:05:53.679137
8	*****	*****	S10	*****	A2A4	2	06-11-2024 10:07:01.475722
9	*****	*****	S10	*****	A3A1	1	06-11-2024 10:08:07.667590
10	*****	*****	S10	*****	A3A2	1	06-11-2024 10:09:13.780087
11	*****	*****	S10	*****	A3A4	1	06-11-2024 10:10:19.441133
12	*****	*****	S10	*****	A4A1	1	06-11-2024 10:11:23.995320
13	*****	*****	S10	*****	A4A2	1	06-11-2024 10:12:29.185896
14	*****	*****	S10	*****	A4A3	2	06-11-2024 10:13:34.006886
15	*****	*****	S10	*****	continue_survey		06-11-2024 10:13:37.173949
16	*****	*****	S10	*****	end		06-11-2024 10:19:47.712318

Figura 20: file Excel in cui sono salvati i timestamp e le scelte relative ad ogni soggetto

Capitolo 3

Elaborazione dei dati

3.1 Pre-processing dati acquisiti

I dati raccolti, da processare in ambiente MATLAB, sono, per ogni soggetto esaminato: TS, muse_data e emotibit_data. Tali dati sono stati quindi importati nel workspace di MATLAB dalle cartelle di origine.

Le fasi di pre-processing hanno l'obiettivo di garantire l'anonimizzazione delle informazioni personali presenti, rendere i dati compatibili con le successive analisi temporali e creare etichette corrispondenti a finestre temporali per ogni audio riprodotto.

Inizialmente, i dati grezzi presenti nei file TS sono stati anonimizzati attraverso l'eliminazione delle colonne contenenti informazioni sensibili, al fine di tutelare la privacy dei partecipanti. Con questo processo è stato possibile lavorare esclusivamente sui dati rilevanti per l'analisi, evitando ogni possibile associazione tra i dati fisiologici e l'identità dei soggetti.

Successivamente, per comprendere la percezione del suono nel mondo reale, è necessario correlare i suoni naturali ai dati EEG; è stata quindi eseguita una segmentazione dei dati raccolti seguendo uno schema ripetitivo, conoscendo a priori le dimensioni delle finestre temporali di ogni audio riprodotto.

Per rendere possibile questa correlazione, i formati temporali presenti nel dataset TS sono stati uniformati attraverso una conversione delle date in timestamp (*epoch*). In questo modo il confronto tra le tempistiche dei dati Muse ed Emotibit e quelle associate agli stimoli sonori, si è reso coerente e preciso.

Il risultato viene salvato aggiungendo una nuova colonna "Epoch" ai dati TS, mostrato in Figura 21.

	1	2	3	4	5	6
	Username	Data di nascita	Audio	Scelta	Timestamp	Epoch
1	'S11'	1998-12-24	'start'	NaN	'07-11-2024 09:51:23.180514'	1.730969483181000e+12
2	'S11'	1998-12-24	'A1A2'	2	'07-11-2024 09:52:29.310132'	1.730969549310000e+12
3	'S11'	1998-12-24	'A1A3'	2	'07-11-2024 09:53:36.415326'	1.730969616415000e+12
4	'S11'	1998-12-24	'A1A4'	2	'07-11-2024 09:54:43.342709'	1.730969683343000e+12
5	'S11'	1998-12-24	'A2A1'	2	'07-11-2024 09:55:48.974359'	1.730969748974000e+12
6	'S11'	1998-12-24	'A2A3'	2	'07-11-2024 09:56:54.272537'	1.730969814273000e+12
7	'S11'	1998-12-24	'A2A4'	2	'07-11-2024 09:57:58.673704'	1.730969878674000e+12
8	'S11'	1998-12-24	'A3A1'	1	'07-11-2024 09:59:03.531175'	1.730969943531000e+12
9	'S11'	1998-12-24	'A3A2'	1	'07-11-2024 10:00:08.458812'	1.730970008459000e+12
10	'S11'	1998-12-24	'A3A4'	1	'07-11-2024 10:01:13.833427'	1.730970073833000e+12
11	'S11'	1998-12-24	'A4A1'	1	'07-11-2024 10:02:18.904583'	1.730970138905000e+12
12	'S11'	1998-12-24	'A4A2'	1	'07-11-2024 10:03:25.361628'	1.730970205362000e+12
13	'S11'	1998-12-24	'A4A3'	1	'07-11-2024 10:04:30.862623'	1.730970270863000e+12
14	'S11'	1998-12-24	'continue_survey'	NaN	'07-11-2024 10:04:35.138950'	1.730970275139000e+12
15	'S11'	1998-12-24	'end'	NaN	'07-11-2024 10:10:45.774678'	1.730970645775000e+12

Figura 21: file TS relativo ai timestamp con aggiunta della colonna "Epoch" tramite elaborazione in MATLAB

Il processing dei dati prevede dapprima la segmentazione dei segnali in base ai timestamp che definiscono inizio e fine di varie fasi, caratterizzate dall'ascolto di suoni differenti.

Per ogni soggetto, i dati relativi al Muse sono stati così elaborati:

- controllo della dimensione del file: è stato definito un valore soglia, fissato al di sopra della media dei file analizzati. Questo criterio è stato adottato per individuare ed escludere file con un numero di dati significativamente ridotto.

Tale procedura è necessaria per preservare l'affidabilità dell'analisi, evitando quindi che file incompleti o poco rappresentativi possano influenzare negativamente i risultati finali.

- I file validi sono stati quindi inseriti all'interno di una struct organizzata in celle ognuna delle quali racchiude i dati relativi ad ogni soggetto.
- Per ciascun file elaborato, i timestamp sono stati confrontati con quelli presenti nel file TS, e sulla base di tale confronto sono state aggiunte tre nuove colonne (vedi Figura 22, Figura 23, Figura 24 e Figura 25):
 - Colonna “Labels1”: questa colonna identifica le diverse fasi del test, fornendo etichette che rappresentano le sequenze temporali previste dalla procedura sperimentale, come descritto nella sezione 2.2. Le etichette utilizzate includono fasi iniziali come “connessione dispositivo”, momenti di silenzio, coppie di stimoli acustici rappresentate da combinazioni quali A1A2, A1A3, A1A4, ecc. Il test si conclude con le fasi etichettate come “sequenza finale” e “disconnessione dispositivo”.
 - Colonna “Labels2”: questa colonna fornisce una descrizione più dettagliata della fase del test, in particolare specifica il tipo di suono associato a ciascun intervallo temporale.
 - Colonna “Scelta”: in questa colonna, per la prima fase del test, vengono riportate le scelte effettuate dai partecipanti, estratte direttamente dal file

TS, in modo da facilitare le analisi successive. Durante gli altri intervalli temporali, la colonna rimane vuota.

La stessa procedura è stata applicata ai dati EmotiBit, tenendo conto delle quattro diverse tipologie di dati raccolti dal sensore. Per organizzare queste informazioni, sono state create quattro strutture separate, una per ciascun tipo di dato, garantendo così una gestione chiara e strutturata dei dati acquisiti.

Questa organizzazione consente una chiara classificazione e segmentazione temporale dei dati, garantendo un'analisi coerente delle diverse fasi temporali.

	1 ts	2 data	3 username	4 pgi	5 pgr	6 pgg	7 Labels1	8 Labels2	9 Scelta
4093	1.730803625454000e+12	05-11-2024 11:47:05.454'S07"		174476	116012	7626"A1A2"		"A1"	"A2"
4094	1.730803625495000e+12	05-11-2024 11:47:05.495'S07"		174469	116035	7640"A1A2"		"A1"	"A2"
4095	1.730803625537000e+12	05-11-2024 11:47:05.537'S07"		174458	116022	7623"A1A2"		"A1"	"A2"
4096	1.730803625595000e+12	05-11-2024 11:47:05.595'S07"		174467	116018	7627"A1A2"		"A1"	"A2"
4097	1.730803625653000e+12	05-11-2024 11:47:05.653'S07"		174449	116023	7616"A1A2"		"A1"	"A2"
4098	1.730803625692000e+12	05-11-2024 11:47:05.692'S07"		174460	116006	7632"A1A2"		"A1"	"A2"
4099	1.730803625731000e+12	05-11-2024 11:47:05.731'S07"		174477	116028	7619"A1A2"		"A1"	"A2"
4100	1.730803625770000e+12	05-11-2024 11:47:05.770'S07"		174478	116028	7633"A1A2"		"A1"	"A2"

Figura 22: etichettatura della prima parte del test

	1 ts	2 data	3 username	4 pgi	5 pgr	6 pgg	7 Labels1	8 Labels2	9 Scelta
5364	1.730803676555000e+12	05-11-2024 11:47:56.555'S07"		173855	116427	7579"A1A2"		"Lag 2 Secondi"	"A2"
5365	1.730803676575000e+12	05-11-2024 11:47:56.575'S07"		173878	116456	7558"A1A2"		"Lag 2 Secondi"	"A2"
5366	1.730803676595000e+12	05-11-2024 11:47:56.595'S07"		173919	116452	7584"A1A2"		"Lag 2 Secondi"	"A2"
5367	1.730803676655000e+12	05-11-2024 11:47:56.655'S07"		173927	116464	7572"A1A2"		"Scelta tra A1A2"	"A2"
5368	1.730803676715000e+12	05-11-2024 11:47:56.715'S07"		173943	116489	7568"A1A2"		"Scelta tra A1A2"	"A2"
5369	1.730803676755000e+12	05-11-2024 11:47:56.755'S07"		173951	116492	7583"A1A2"		"Scelta tra A1A2"	"A2"
5370	1.730803676795000e+12	05-11-2024 11:47:56.795'S07"		173983	116515	7569"A1A2"		"Scelta tra A1A2"	"A2"

Figura 23: etichettatura della fase di scelta nella prima parte del test

	1 ts	2 data	3 username	4 pgi	5 pgr	6 pgg	7 Labels1	8 Labels2	9 Scelta
24875	1.730804460634000e+12	05-11-2024 12:01:00.634'S07"		175397	119569	7880"Sequenza Finale"		"Silenzio 2"	"
24876	1.730804460655000e+12	05-11-2024 12:01:00.655'S07"		175391	119560	7893"Sequenza Finale"		"Silenzio 2"	"
24877	1.730804460675000e+12	05-11-2024 12:01:00.675'S07"		175397	119551	7870"Sequenza Finale"		"Silenzio 2"	"
24878	1.730804460735000e+12	05-11-2024 12:01:00.735'S07"		175401	119548	7890"Sequenza Finale"		"Ruscello"	"
24879	1.730804460794000e+12	05-11-2024 12:01:00.794'S07"		175391	119548	7881"Sequenza Finale"		"Ruscello"	"
24880	1.730804460815000e+12	05-11-2024 12:01:00.815'S07"		175391	119549	7883"Sequenza Finale"		"Ruscello"	"

Figura 24: etichettatura della seconda parte del test

	1	2	3	4	5	6	7	8	9
	ts	data	username	pgi	pgr	pgg	Labels1	Labels2	Scelta
28109	1.730804590546000e+12	05-11-2024 12:03:10.546	'S07'	176199	119801	7956	"Sequenza Finale"	"Silenzio 3"	""
28110	1.730804590576000e+12	05-11-2024 12:03:10.576	'S07'	176220	119801	7980	"Sequenza Finale"	"Silenzio 3"	""
28111	1.730804590636000e+12	05-11-2024 12:03:10.636	'S07'	176202	119798	7976	"Sequenza Finale"	"Silenzio 3"	""
28112	1.730804590695000e+12	05-11-2024 12:03:10.695	'S07'	176226	119801	7968	"Sequenza Finale"	"Silenzio 3"	""
28113	1.730804590754000e+12	05-11-2024 12:03:10.754	'S07'	176198	119797	7975	"Sequenza Finale"	"Fastidioso"	""
28114	1.730804590785000e+12	05-11-2024 12:03:10.785	'S07'	176226	119809	7976	"Sequenza Finale"	"Fastidioso"	""
28115	1.730804590815000e+12	05-11-2024 12:03:10.815	'S07'	176225	119803	7981	"Sequenza Finale"	"Fastidioso"	""

Figura 25: etichettatura della seconda parte del test

3.2 Pre-processing dati EEG

Una volta effettuata la segmentazione temporale dei dati, è possibile procedere con la fase di filtraggio del segnale EEG, essenziale per rimuovere artefatti e rumori indesiderati. I dati Muse del segnale EEG sono stati processati utilizzando un codice scritto in MATLAB il quale implementa una serie di passaggi chiave per garantire l'accuratezza dell'elaborazione e l'idoneità dei segnali per le analisi successive.

3.2.1 Filtraggio dati EEG

I segnali EEG analizzati si riferiscono alla seconda fase del test, includendo il silenzio di intervallo tra la prima e la seconda fase, nonché la prima riproduzione della sequenza di stimoli acustici, costituita da un suono piacevole e uno fastidioso.

I dati EEG sono stati acquisiti con una frequenza di campionamento di 256 Hz ed è stato implementato un filtro Butterworth del quarto ordine, con banda passante tra 0.5 e 42 Hz, che permette di eliminare le componenti associate a interferenze muscolari o ambientali. Dopo il filtraggio, è stata applicata la funzione *detrend* per rimuovere eventuali tendenze lineari del segnale. Tali passaggi sono riportati nell'Appendice B.

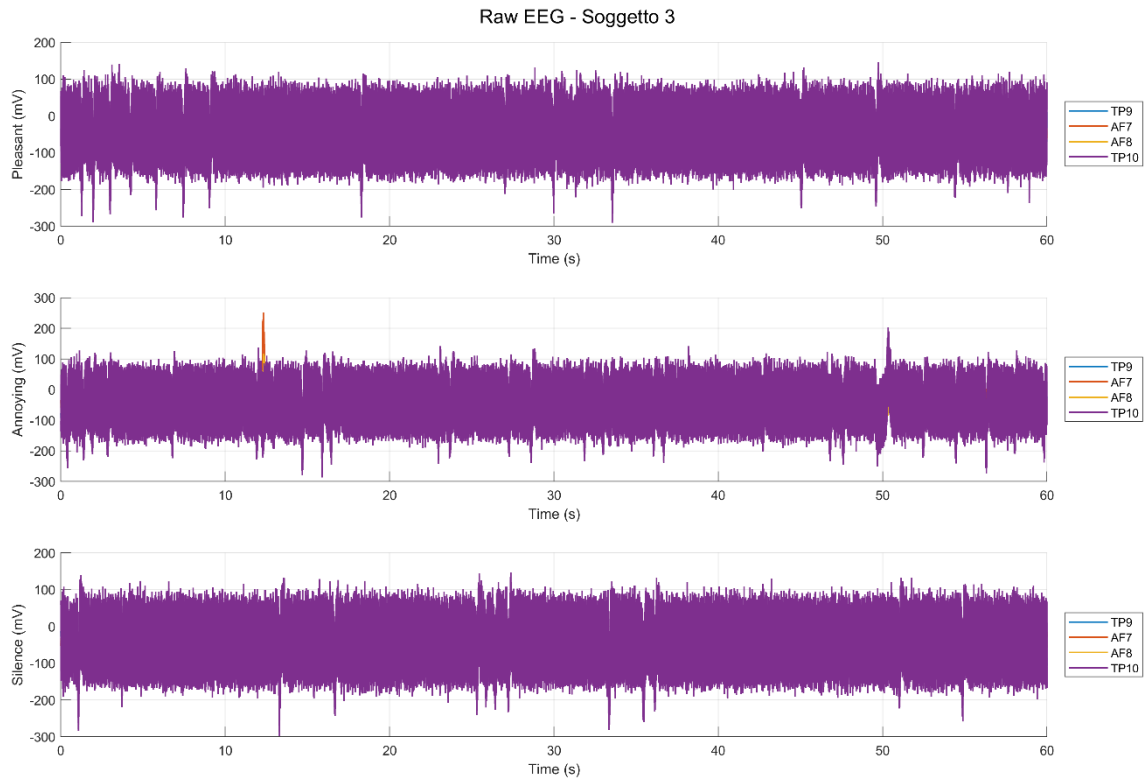


Figura 26: segnale EEG del soggetto 3 nel dominio del tempo, prima dell'applicazione del filtraggio

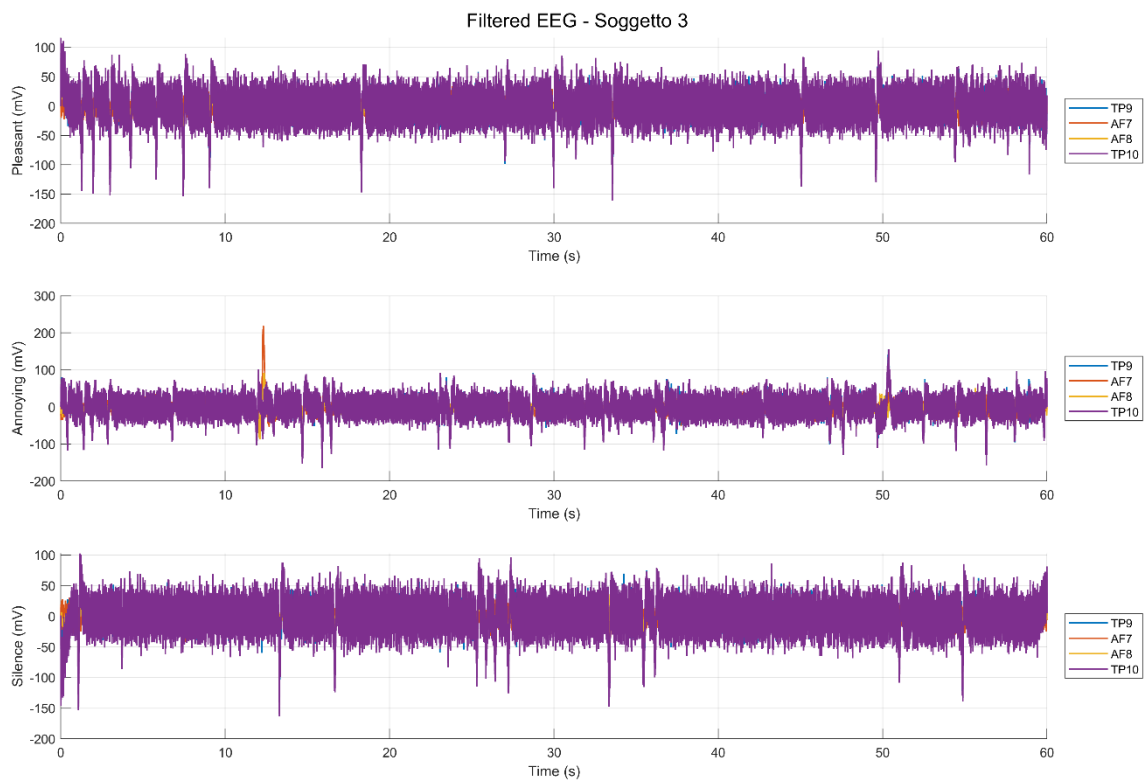


Figura 27: segnale EEG del soggetto 3 nel dominio del tempo, dopo l'applicazione del filtraggio

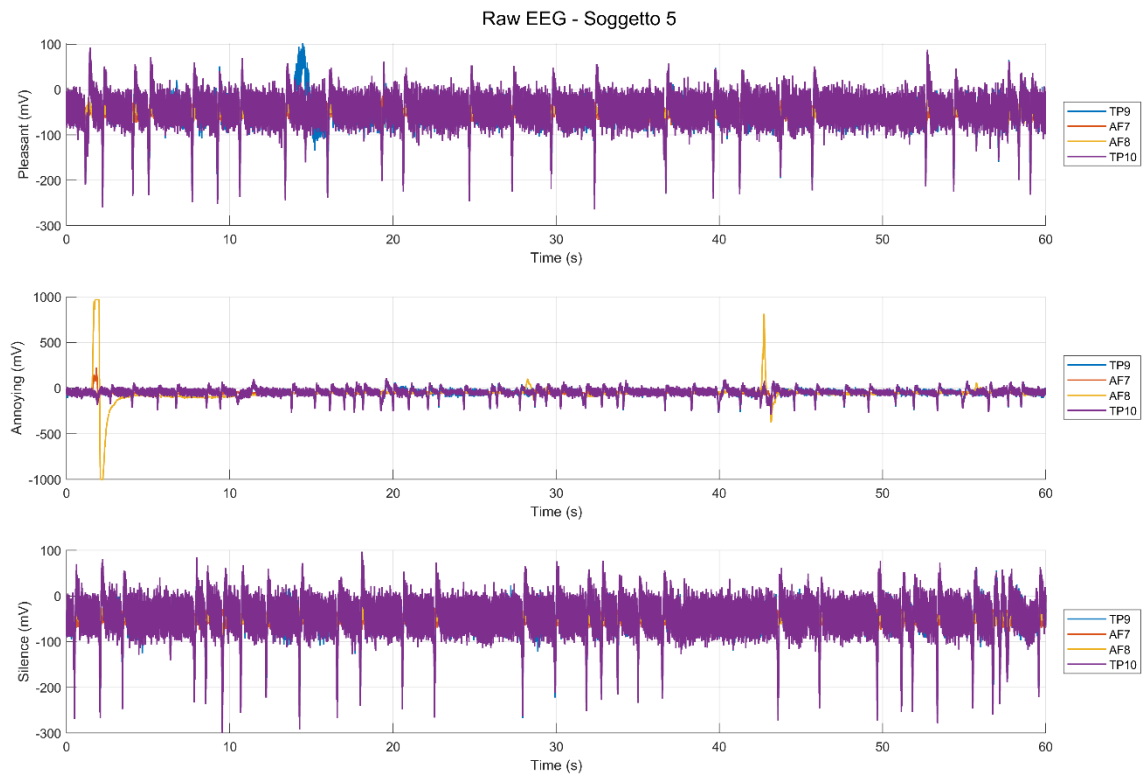


Figura 28: segnale EEG del soggetto 5 nel dominio del tempo, prima dell'applicazione del filtraggio

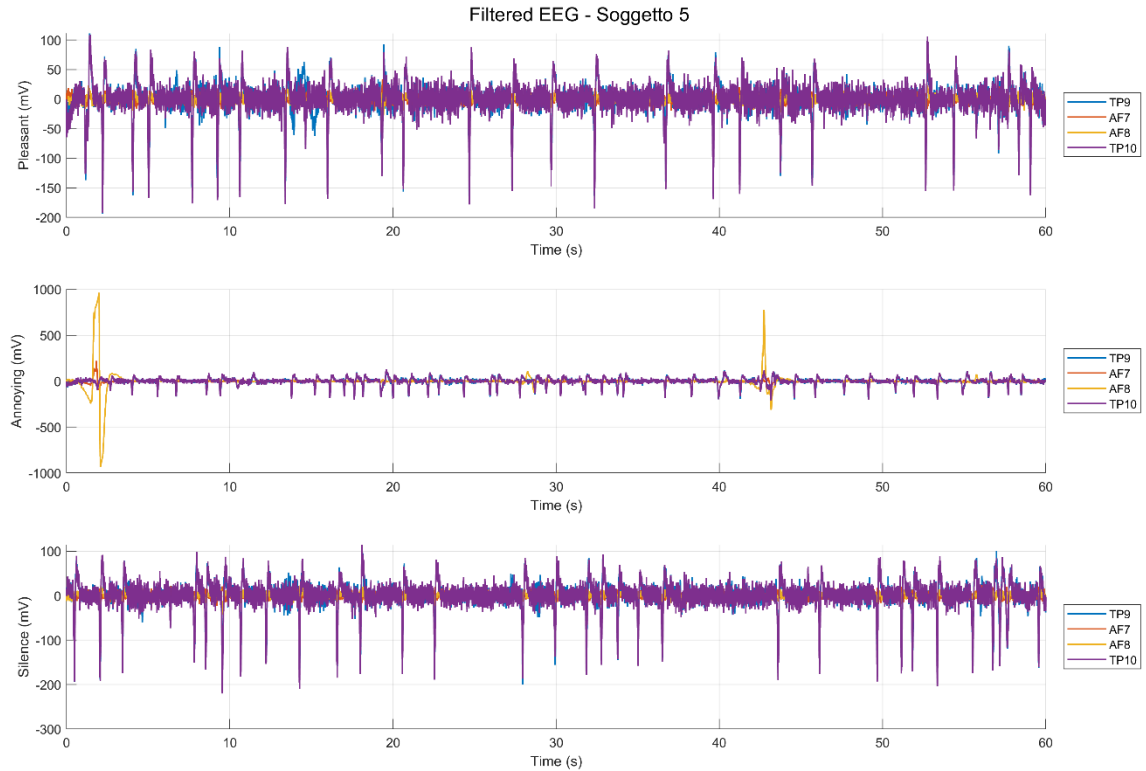


Figura 29: segnale EEG del soggetto 5 nel dominio del tempo, dopo l'applicazione del filtraggio

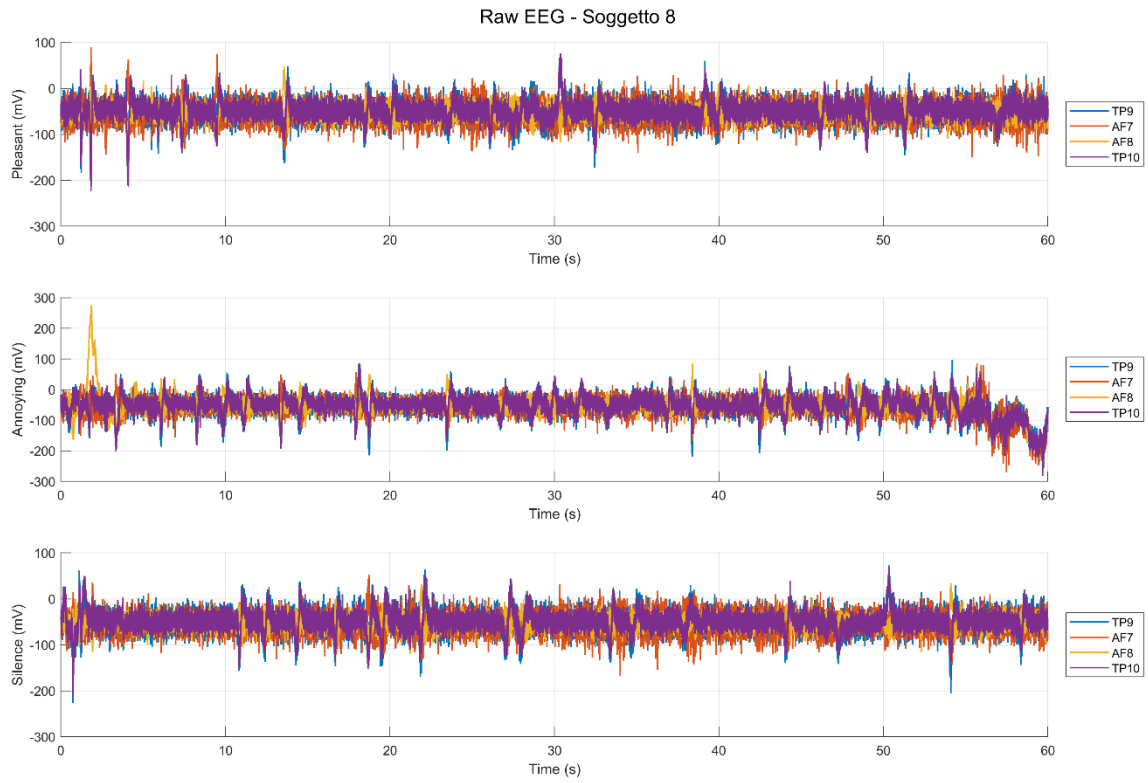


Figura 30: segnale EEG del soggetto 8 nel dominio del tempo, prima dell'applicazione del filtraggio

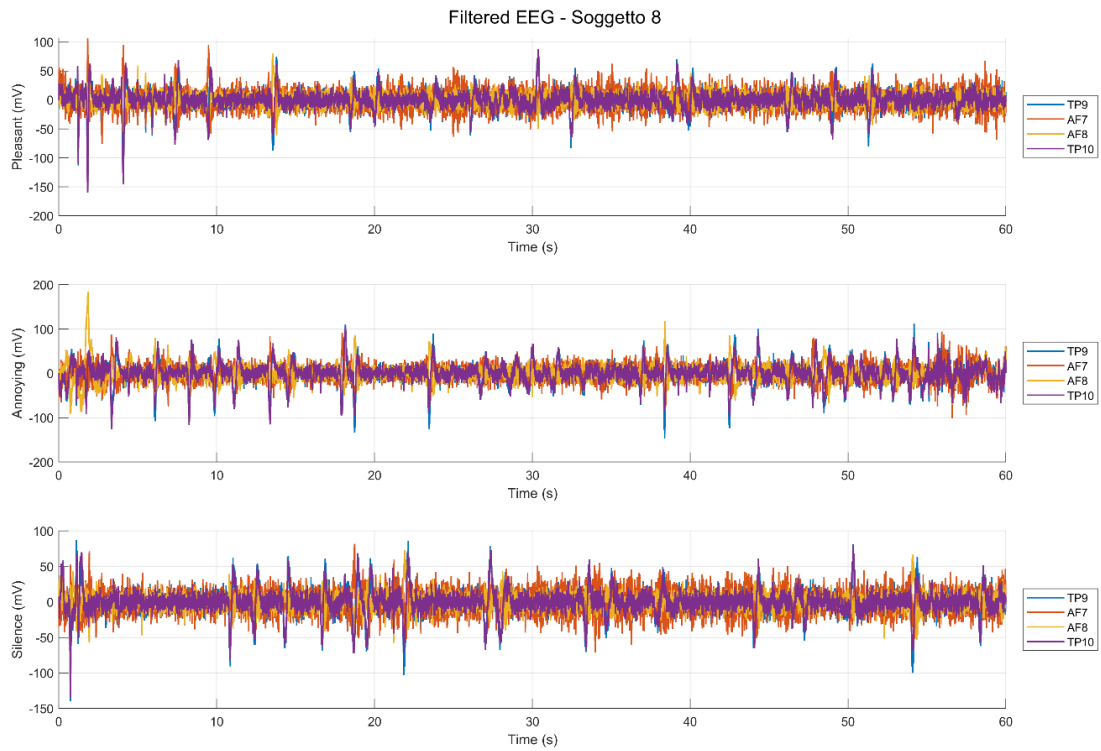


Figura 31: segnale EEG del soggetto 8 nel dominio del tempo, dopo l'applicazione del filtraggio

Le storie temporali dei segnali EEG, riportate dalla Figura 26 alla Figura 31 per 3 soggetti, forniscono informazioni sulla qualità dell'acquisizione e consentono di individuare eventuali problematiche legate al rumore o ad artefatti. Nonostante l'utilizzo dell'app Muse per il controllo dell'impedenza degli elettrodi, sono stati riscontrati errori nell'acquisizione del segnale EEG, che hanno compromesso la qualità dei dati per alcuni soggetti.

Dall'analisi preliminare delle storie temporali, è emerso che alcuni segnali presentano anomalie significative. In particolare, nella Figura 26 e nella Figura 27 si nota un segnale apparentemente saturato, con ampiezze atipiche e un andamento che potrebbe indicare la presenza di artefatti o distorsioni nel segnale EEG. Anche dopo il filtraggio, il segnale risulta ancora chiaramente distorto, suggerendo che il problema non sia riconducibile esclusivamente al rumore di fondo, ma probabilmente ad errori di acquisizione o a interferenze strumentali.

Poiché l'identificazione di segnali EEG di scarsa qualità è fondamentale per garantire l'affidabilità delle analisi, oltre all'eliminazione dei dati dei soggetti con dimensione limitata (come descritto nella sezione 3.1), si propone di integrare un'ulteriore fase di selezione, che preveda l'eliminazione a posteriori dei soggetti per cui è visibile un'errata acquisizione EEG. Questa strategia permette di escludere dal dataset i segnali EEG che presentano distorsioni evidenti o livelli di rumore incompatibili con un'analisi accurata, evitando di introdurre errori sistematici nelle analisi successive.

Nei segnali validi, l'analisi delle storie temporali relative alle condizioni di silenzio, suono piacevole e suono fastidioso, evidenzia un miglioramento della qualità del segnale a seguito del filtraggio. L'attenuazione del rumore di fondo ha permesso di ridurre le

interferenze indesiderate, preservando al contempo la struttura delle componenti EEG di interesse. Questa operazione ha reso più chiara l'interpretazione del segnale, facilitando l'analisi delle risposte cerebrali ai diversi stimoli sonori.

3.3 Analisi PSD

La densità spettrale di potenza (PSD) è una misura che descrive come l'energia di un segnale si distribuisce nelle diverse frequenze. Nel caso dei segnali EEG, consente di analizzare la presenza e l'intensità delle varie bande di frequenza associate all'attività cerebrale, come delta, theta, alpha e beta. Questo tipo di analisi è particolarmente utile per confrontare le risposte neurali a diversi stimoli e per individuare eventuali variazioni tra le condizioni sperimentali considerate.

La densità spettrale di potenza è stata stimata utilizzando il metodo di Welch, implementato tramite la funzione *pwelch* di MATLAB. Questo metodo migliora la stima della PSD rispetto alla trasformata di Fourier diretta, riducendo il rumore e la varianza dell'output grazie alla suddivisione del segnale in segmenti sovrapposti. Nello specifico, il segnale EEG è stato suddiviso in finestre di Hanning della durata di 0.5 secondi con una sovrapposizione del 50%. Questo approccio permette di ottenere una stima più stabile della distribuzione spettrale dell'energia, attenuando le fluttuazioni dovute alla limitata estensione temporale del segnale. Nell'Appendice B sono illustrati tali passaggi.

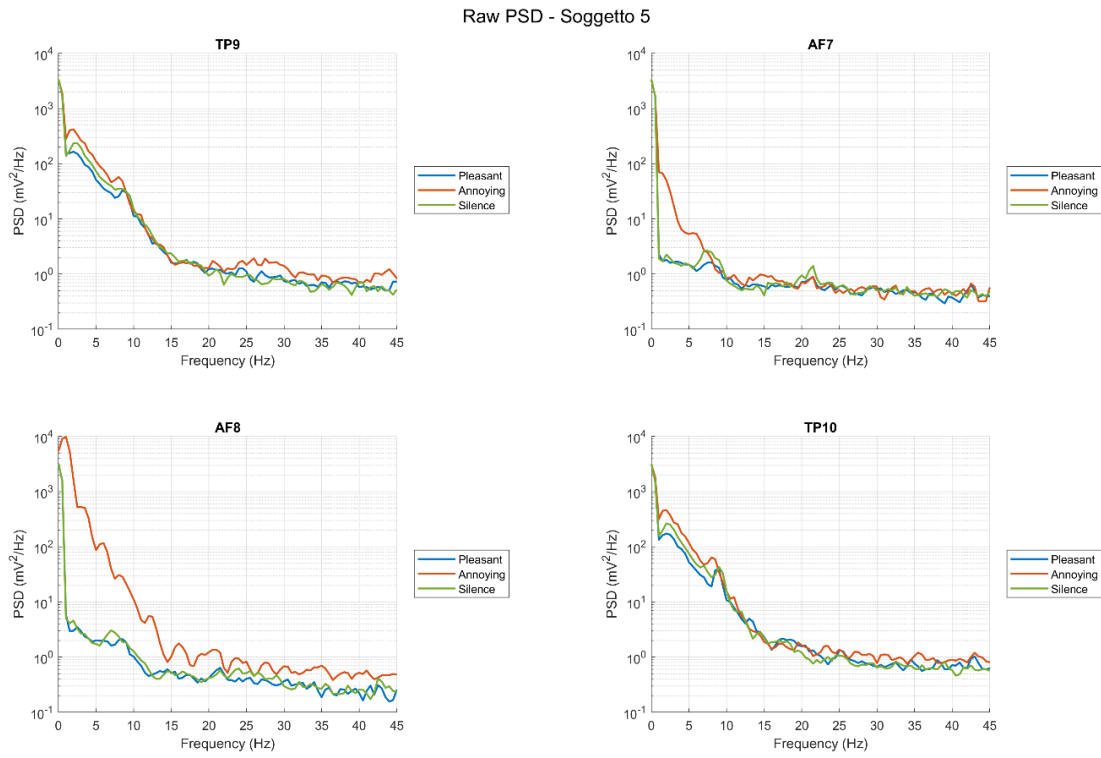


Figura 32: andamento della densità spettrale di potenza (PSD) per il soggetto 5 nei quattro canali EEG, prima del filtraggio

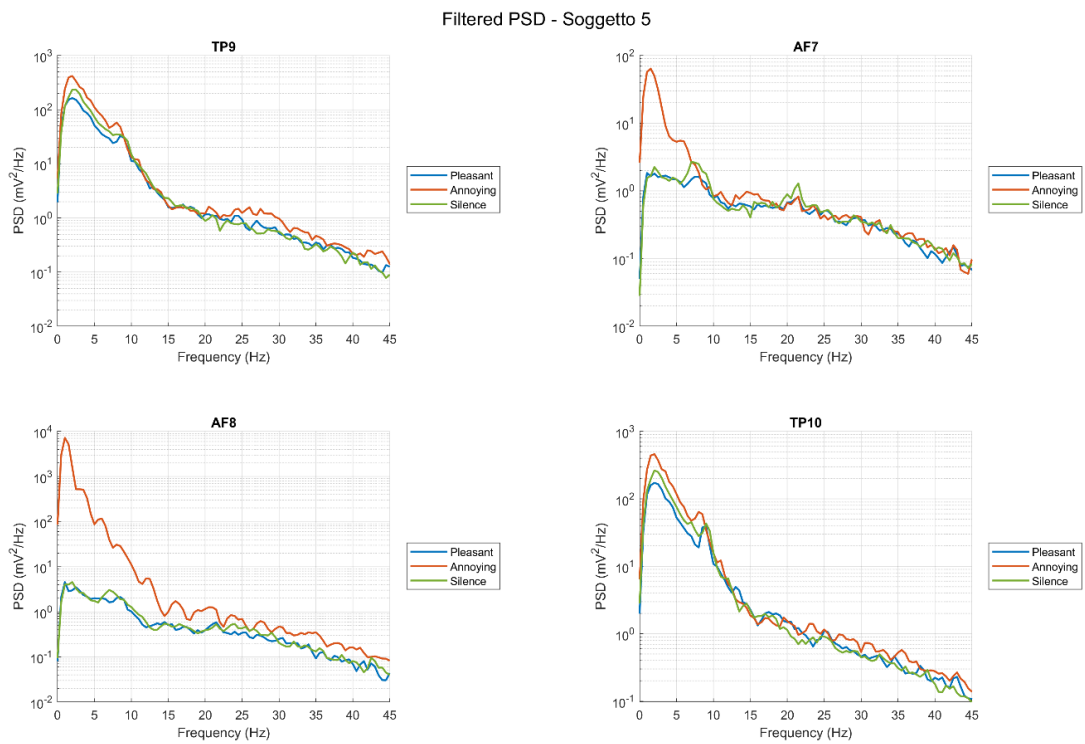


Figura 33: andamento della densità spettrale di potenza (PSD) per il soggetto 5 nei quattro canali EEG, dopo il filtraggio

Raw PSD - Soggetto 8

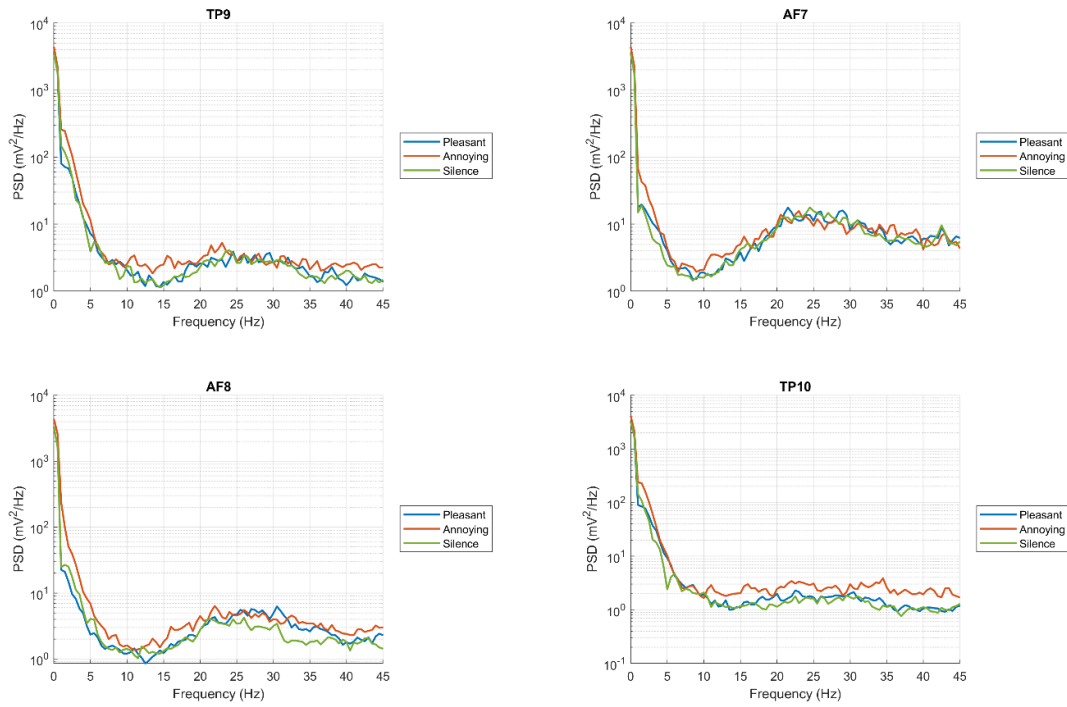


Figura 34: andamento della densità spettrale di potenza (PSD) per il soggetto 8 nei quattro canali EEG, prima del filtraggio

Filtered PSD - Soggetto 8

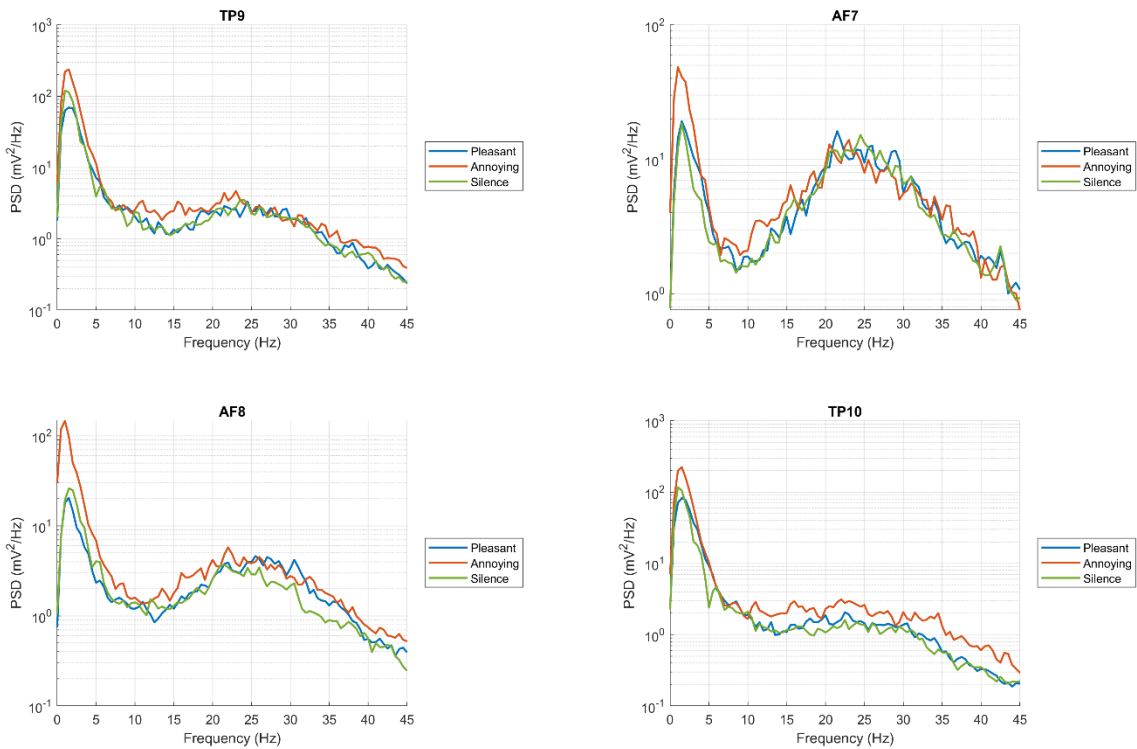


Figura 35: andamento della densità spettrale di potenza (PSD) per il soggetto 8 nei quattro canali EEG, dopo il filtraggio

Il confronto tra le densità spettrali di potenza (PSD) dei segnali grezzi e filtrati, osservabile dalla Figura 32 alla Figura 35, evidenzia una significativa attenuazione delle componenti a bassa frequenza (<5 Hz), generalmente associate a rumore e artefatti, senza alterare la distribuzione delle bande di frequenza di interesse (theta, alpha, beta). Dopo il filtraggio, il segnale risulta più definito e le differenze tra le condizioni sperimentali (pleasant, annoying, silence) appaiono più evidenti, rendendo il dato più affidabile per l'analisi delle risposte cerebrali agli stimoli sonori.

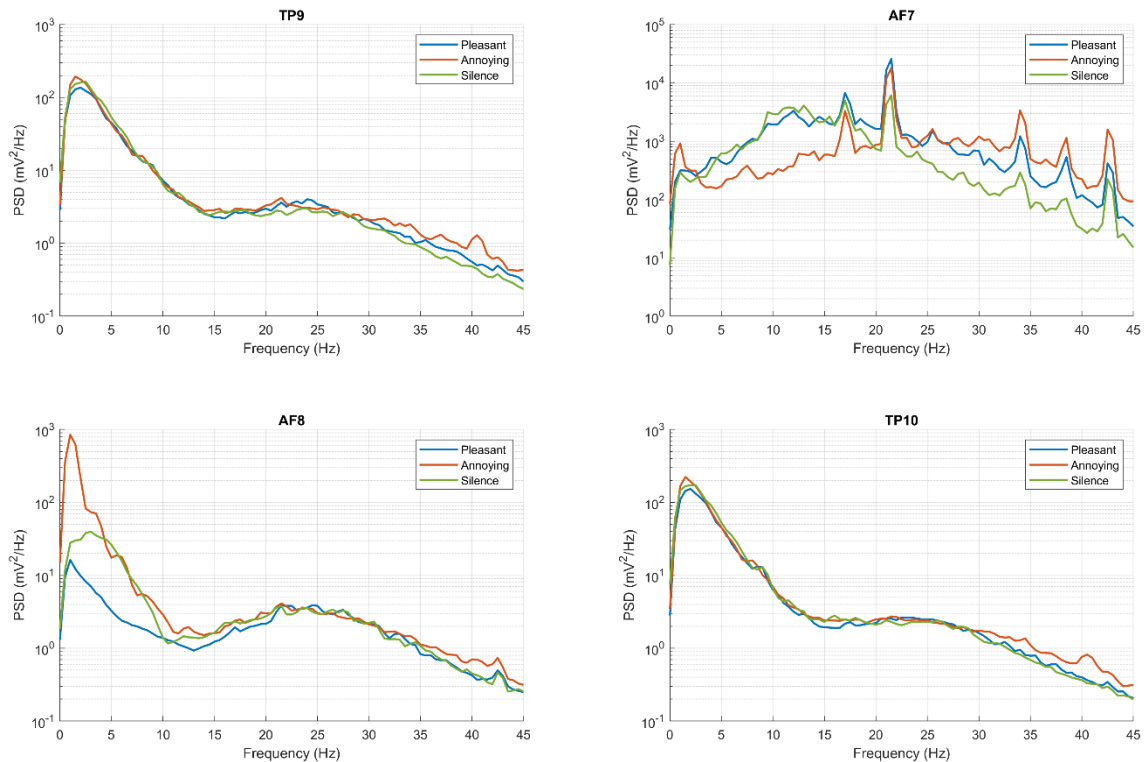


Figura 36: PSD filtrata mediata su tutti i soggetti nei quattro canali EEG, per confrontare le risposte spettrali alle diverse condizioni sperimentali (suono piacevole, suono fastidioso, silenzio)

Il confronto tra le PSD dei singoli soggetti e la media complessiva, Figura 36, evidenzia una certa variabilità interindividuale nella risposta EEG agli stimoli sonori. Tuttavia,

analizzando l'andamento medio, emergono alcune tendenze ricorrenti. Nei singoli soggetti, queste differenze risultano più o meno marcate a seconda della variabilità individuale, suggerendo una risposta EEG soggettiva agli stimoli acustici. La media complessiva, pur mantenendo queste caratteristiche generali, attenua le variazioni estreme osservate nei singoli individui, restituendo un quadro d'insieme più stabile e rappresentativo dell'andamento globale.

Capitolo 4

Conclusioni

Lo studio presentato in questa tesi ha esplorato la correlazione tra la percezione soggettiva degli stimoli acustici e le risposte fisiologiche, con particolare attenzione ai segnali EEG. L'integrazione dei Jury test acustici con l'analisi neurofisiologica ha permesso di indagare come i suoni percepiti piacevoli o fastidiosi influenzino l'attività cerebrale fornendo un approccio oggettivo per la valutazione dell'esperienza.

Dal punto di vista metodologico, il pre-processing dei dati Muse ed EmotiBit è stato applicato a tutti i segnali raccolti, assicurando la corretta sincronizzazione con i timestamp associati agli stimoli acustici. Il filtraggio del segnale EEG è stato effettuato solo relativamente alla seconda parte del test, con l'obiettivo di ridurre il rumore senza alterare le caratteristiche principali dell'attività cerebrale. Per l'analisi, è stata utilizzata la densità spettrale di potenza (PSD), parametro utile per valutare le variazioni dell'attività neurale in risposta ai suoni presentati.

I risultati ottenuti indicano che combinare dati soggettivi e fisiologici può essere una strategia efficace per studiare la percezione acustica e il modo in cui il cervello reagisce agli stimoli sonori. Tuttavia, alcune criticità restano da considerare, in particolare la possibile presenza di artefatti nel segnale EEG, dovuti all'uso di dispositivi portatili, e la variabilità individuale nella percezione dei suoni.

Uno degli svantaggi principali del test di confronto a coppie è il tempo di esecuzione che aumenta esponenzialmente con l'aggiunta di ogni nuovo suono da valutare: ogni

confronto prevede la riproduzione di due suoni, senza la possibilità di saltarne l'ascolto dopo pochi secondi. Per garantire l'affidabilità dei risultati è opportuno implementare un controllo di coerenza: un giurato coerente, avendo ascoltato due volte la stessa coppia di suoni, dovrebbe selezionare sempre lo stesso suono come preferito all'interno della coppia. Tuttavia, questa verifica non è stata effettuata nel contesto di questa analisi.

Inoltre, per una comprensione più approfondita dei segnali prelevati nella seconda fase del test, sarebbe opportuno effettuare analisi più articolate in esperimenti futuri. Ad esempio, una valida alternativa potrebbe essere una valutazione su scala da parte dei soggetti, per quantificare il grado di piacevolezza percepita degli stimoli acustici. Questo approccio potrebbe rendere possibile un confronto tra la percezione soggettiva, espressa tramite la valutazione, e la percezione oggettiva, ovvero quella rilevata dai segnali fisiologici.

Analisi future potrebbero migliorare significativamente la comprensione della correlazione tra i segnali fisiologici e le risposte dei giurati, permettendo di identificare con maggiore precisione gli stati emotivi suscitati nei soggetti.

Inoltre, è importante considerare che i dispositivi wireless utilizzati per la raccolta dei dati possono essere soggetti a perdite di informazioni a causa della riduzione della batteria.

In conclusione, questa analisi, seppur preliminare, offre indicazioni utili e apre la strada ad ulteriori approfondimenti. Viene evidenziata quindi la necessità di affinare gli aspetti metodologici e statistici per una valutazione più completa e affidabile dei risultati.

Appendice A

```
clear all;
close all;
clc;

% Cartelle contenenti i file
folderPathTS = 'C:\Users\nucci\Desktop\tirocinio\codice\TS';
folderPathMUSE = 'C:\Users\nucci\Desktop\tirocinio\codice\muse_data';
folderPathEMOTIBIT = 'C:\Users\nucci\Desktop\tirocinio\codice\emotibit_data';

% Pattern per individuare i file CSV
filePatternTS = fullfile(folderPathTS, '*.csv');
fileTSList = dir(filePatternTS);

filePatternMUSE = fullfile(folderPathMUSE, '*.csv');
fileMUSEList = dir(filePatternMUSE);

% Definire le combinazioni cicliche
combinazioni = ["A1A2", "A1A3", "A1A4", "A2A1", "A2A3", "A2A4", "A3A1", "A3A2",
"A3A4", "A4A1", "A4A2", "A4A3"];
```

Elaborazione file TS

Controllo ed elaborazione iniziale dei file TS

```
for i = 1:length(fileTSList)

    fileName_TS = fullfile(folderPathTS, fileTSList(i).name);
    dataTS = readtable(fileName_TS, 'VariableNamingRule', 'preserve');

    % Rimozione colonne nome e cognome
    dataTS(:, 1:2) = [];

    % Controllo che il file TS non sia vuoto
    if isempty(dataTS) || size(dataTS, 2) == 0
        fprintf('Errore: il file "%s" non contiene dati validi.\n',
fileTSList(i).name);
        continue; % Passa al prossimo file
    end

    % Individua ultima colonna con data
    lastColumn = size(dataTS, 2);
    if all(cellfun(@isempty, table2cell(dataTS(:, lastColumn)) ))
        fprintf('Errore: la colonna timestamp nel file "%s" è vuota.\n',
fileTSList(i).name);
        continue; % Passa al prossimo file
    end

    % Convertito Timestamp
    timestampColumn = string(dataTS(:, lastColumn));
```

```

formatoInput = 'dd-MM-yyyy HH:mm:ss.SSSSSS';
dateTimes = datetime(timestampColumn, 'InputFormat', formatoInput, 'TimeZone',
'Europe/Rome');
epochTimes = round(posixtime(dateTimes) * 1000, 0);

% Aggiunge gli epoch al file TS
dataTS_unix{i} = addvars(dataTS, epochTimes, 'NewVariableNames', 'Epoch');

% Sovrascrivere il file originale
% writetable(dataTS, fileName_TS, 'Delimiter', ';', 'WriteVariableNames', true);
end

```

Elaborazione ed etichettatura dei dati Muse

```

DATA_MUSE = struct();

for i = 1: length (fileMUSEList)

    sogg = fileMUSEList(i).name;
    sogg_valid = strrep (sogg, '.', '_');

    epochTS = dataTS_unix{:,i}.Epoch;
    sceltaTS = dataTS_unix{i}.Scelta(2:13);

    % Leggere il file Muse
    fileName_MUSE = fullfile(folderPathMUSE, fileMUSEList(i).name);
    dataMUSE = readtable(fileName_MUSE, 'VariableNamingRule', 'preserve');

    % Uniformare lunghezza timestamp a 13 cifre
    epochMUSE = floor(dataMUSE.ts * 1000); %variabile timestamp troncati a 13 cifre
    dataMUSE.ts = epochMUSE; % sovrascrivo timestamp a lunghezze variabili con ts a
    13 cifre per tutti

    % conversione epoch in data
    dataMUSEformattata = datetime(dataMUSE.ts / 1000, 'ConvertFrom', 'posixtime',
'Format', 'dd-MM-yyyy HH:mm:ss.SSS', 'TimeZone', 'Europe/Rome');

    % Aggiungere la colonna 'data' nella seconda posizione
    dataMUSE.data = dataMUSEformattata;
    dataMUSE = dataMUSE(:, [1, end, 2:end-1]);

    % Calcolare gli intervalli di tempo in millisecondi (RECUPERO DEI SILENZI
    INIZIALI)
    j=1;
    timestamp1 = epochTS(j); % Primo timestamp del file TS
    timestamp1meno30 = timestamp1 - 30 * 1000; % Sottrae 30 secondi in millisecondi

    % Intervalli
    intervalloSilenzio1Min = timestamp1meno30 - 20 * 1000; % timestamp1meno30 - 20
    secondi in millisecondi

```

```

    intervalloSilenzio1Max = timestamp1meno30 + 10 * 1000; % timestamp1meno30 + 10
    secondi in millisecondi

    % Inizializzare le etichette
    dataMUSE.Labels1 = strings(height(dataMUSE), 1);
    dataMUSE.Labels2 = strings(height(dataMUSE), 1);
    dataMUSE.Scelta = strings(height(dataMUSE), 1);

    % Applico le etichette LABELS 1 ai dati Muse
    dataMUSE.Labels1(epochMUSE < intervalloSilenzio1Min) = "Connessione
Dispositivo";
    dataMUSE.Labels1(epochMUSE >= intervalloSilenzio1Min & epochMUSE <=
intervalloSilenzio1Max) = "Silenzio 1";
    dataMUSE.Labels1(epochMUSE >= intervalloSilenzio1Max & epochMUSE <= timestamp1)
= "Scarto Pre-inizio Test";

    % Applico le etichette LABELS 2 ai dati Muse
    dataMUSE.Labels2(epochMUSE < intervalloSilenzio1Min) = "Connessione
Dispositivo";
    dataMUSE.Labels2(epochMUSE >= intervalloSilenzio1Min & epochMUSE <=
intervalloSilenzio1Max) = "Silenzio 1";
    dataMUSE.Labels2(epochMUSE >= intervalloSilenzio1Max & epochMUSE <= timestamp1)
= "Scarto Pre-inizio Test";

    % Iterare per ogni timestamp a partire dal secondo
    for j = 1:12
        currentTimestamp = epochTS(j);
        nextTimestamp = epochTS(j + 1);

        % Calcolare i limiti degli intervalli
        intervalA1_start = currentTimestamp; % Inizio primo
        audio della coppia
        intervalA1_end = currentTimestamp + 30 * 1000; % +30 secondi
        intervallag1_start = intervalA1_end; % Fine primo
        audio della coppia
        intervallag1_end = intervallag1_start + 2 * 1000; % +2 secondi
        intervalA2_start = intervallag1_end; % Inizio
        secondo audio della coppia
        intervalA2_end = intervalA2_start + 30 * 1000; % +30 secondi
        intervallag2_start = intervalA2_end; % Fine secondo
        audio della coppia
        intervallag2_end = intervallag2_start + 2 * 1000; % +2 secondi
        intervalChoice_start = intervallag2_end; % Inizio scelta
        intervalChoice_end = nextTimestamp; % Fine scelta
        (prossimo timestamp)

        % Determinare l'indice ciclico per la combinazione di scelte
        % Determinare combinazione corrente
        combinazioneCorrente = combinazioni(j);

```

```

    % Applicare le etichette LABELS 1 ai dati Muse
    dataMUSE.Labels1(epochMUSE >= intervalA1_start & epochMUSE <
intervalChoice_end) = combinazioneCorrente;

    % Applicare le etichette LABELS 2! ai dati Muse
    dataMUSE.Labels2(epochMUSE >= intervalA1_start & epochMUSE <
intervalA1_end) = extractBetween(combinazioneCorrente, 1, 2);
    dataMUSE.Labels2(epochMUSE >= intervalLag1_start & epochMUSE <
intervallag1_end) = "Lag 2 Secondi";
    dataMUSE.Labels2(epochMUSE >= intervalA2_start & epochMUSE <
intervalA2_end) = extractBetween(combinazioneCorrente, 3, 4);
    dataMUSE.Labels2(epochMUSE >= intervalLag2_start & epochMUSE <
intervallag2_end) = "Lag 2 Secondi";
    dataMUSE.Labels2(epochMUSE >= intervalChoice_start & epochMUSE <
intervalChoice_end) = "Scelta tra " + combinazioneCorrente;

    opzione1 = extractBetween(combinazioneCorrente, 1, 2); % estrae primo suono
    opzione2 = extractBetween(combinazioneCorrente, 3, 4); % estrae secondo
suono

    % Determinare la scelta associata
    sceltaCorrente = sceltaTS(j);
    if sceltaCorrente == 1
        sceltaAssegnata = opzione1; % Scelta 1 corrisponde a opzione1
    elseif sceltaCorrente == 2
        sceltaAssegnata = opzione2; % Scelta 2 corrisponde a opzione2
    else
        sceltaAssegnata = "Errore"; % Caso di errore
    end

    dataMUSE.Scelta(epochMUSE >= intervalA1_start & epochMUSE <
intervalChoice_end) = sceltaAssegnata;
end

j=13;
currentTimestamp = epochTS(j);
nextTimestamp = epochTS(j + 1);
% Applico le etichette LABELS 1 ai dati Muse
dataMUSE.Labels1(epochMUSE >= currentTimestamp & epochMUSE <= nextTimestamp) =
"Ancora qualche minuto...";
% Applico le etichette LABELS 2 ai dati Muse
dataMUSE.Labels2(epochMUSE >= currentTimestamp & epochMUSE <= nextTimestamp) =
"Ancora qualche minuto...";

j=14;
currentTimestamp = epochTS(j);
nextTimestamp = epochTS(j + 1);
ts_fine = dataMUSE(end,1);

```

```

ts_fine1 = table2array(ts_fine);

% Applico le etichette LABELS 1 ai dati Muse
dataMUSE.Labels1(epochMUSE >= currentTimestamp & epochMUSE <= nextTimestamp) =
"Sequenza Finale";
dataMUSE.Labels1(epochMUSE >= nextTimestamp & epochMUSE <= ts_fine1) =
"Disconnessione Dispositivi";

% Applico le etichette LABELS 2 ai dati Muse
dataMUSE.Labels2(epochMUSE >= nextTimestamp & epochMUSE <= ts_fine1) =
"Disconnessione Dispositivi";

% Calcolare intervalli temporali della sequenza finale
silenzio2_start = currentTimestamp;           % Inizio sequenza finale
silenzio2_end = silenzio2_start + 60 * 1000;   % Fine silenzio 2
ruscello_start = silenzio2_end;               % Inizio ruscello 1
ruscello_end = ruscello_start + 60 * 1000;     % Fine ruscello 1
fastidioso1_start = ruscello_end;            % Inizio fastidioso 1
fastidioso1_end = fastidioso1_start + 60 * 1000; % Fine fastidioso 1
silenzio3_start = fastidioso1_end;           % Inizio silenzio 3
silenzio3_end = silenzio3_start + 10 * 1000;   % Fine silenzio 3
fastidioso2_start = silenzio3_end;           % Inizio fastidioso 2
fastidioso2_end = fastidioso2_start + 60 * 1000; % Fine fastidioso 2
ruscello2_start = fastidioso2_end;           % Inizio ruscello 2
ruscello2_end = ruscello2_start + 60 * 1000;   % Fine ruscello2
silenzio4_start = ruscello2_end;              % Inizio silenzio 4
silenzio4_end = nextTimestamp;                % Fine silenzio 4 (fine
test)

dataMUSE.Labels2(epochMUSE >= silenzio2_start & epochMUSE < silenzio2_end) =
"Silenzio 2";
dataMUSE.Labels2(epochMUSE >= ruscello_start & epochMUSE < ruscello_end) =
"Ruscello";
dataMUSE.Labels2(epochMUSE >= fastidioso1_start & epochMUSE < fastidioso1_end)
= "Fastidioso";
dataMUSE.Labels2(epochMUSE >= silenzio3_start & epochMUSE < silenzio3_end) =
"Silenzio 3";
dataMUSE.Labels2(epochMUSE >= fastidioso2_start & epochMUSE < fastidioso2_end)
= "Fastidioso";
dataMUSE.Labels2(epochMUSE >= ruscello2_start & epochMUSE < ruscello2_end) =
"Ruscello";
dataMUSE.Labels2(epochMUSE >= silenzio4_start & epochMUSE < silenzio4_end) =
"Silenzio 4";

DATA_MUSE.(sogg_valid) = dataMUSE;

% Sovrascrivere il file MUSE con le etichette
% writetable(dataMUSE, fileName_MUSE, 'Delimiter', ';', 'WriteVariableNames',
true);
end

```

Rimozione file Muse non validi

```
for i = 1 : length(fileMUSEList)
    sogg = fileMUSEList(i).name;
    sogg_valid = strrep(sogg, '.', '_'); %sostituisco _ con .
    filePathMUSE = fullfile(folderPathMUSE, sogg);

    % Controllo dimensione dei dati MUSE
    if isfield(DATA_MUSE, sogg_valid) && ~isempty(DATA_MUSE.(sogg_valid))
        if ~isempty(filePathMUSE)
            MUSEInfo = dir(filePathMUSE);
            fileSizeKB = MUSEInfo.bytes / 1024;
            if fileSizeKB <= 18000
                fprintf('ERRORE: File MUSE per %s ha dimensione %.2f KB - Non
supera 20000 KB\n', sogg_valid, fileSizeKB);
                DATA_MUSE.(sogg_valid) = [];
            end
        else
            fprintf('ERRORE: Nessun file MUSE trovato per %s\n', sogg_valid);
            DATA_MUSE.(sogg_valid) = [];
        end
    end
end
```

Elaborazione ed etichettatura dei dati Emotibit

```
fileEMOTIBITList = dir(folderPathEMOTIBIT);
fileEMOTIBITList = fileEMOTIBITList([fileEMOTIBITList.isdir] &
~ismember({fileEMOTIBITList.name}, {'.', '..'})); % Solo sottocartelle valide

DATA_EDA = struct();
DATA_PPG = struct();
DATA_MOTION = struct();
DATA_TEMPERATURE = struct();

for i = 1:length(fileEMOTIBITList)
    sogg = fileEMOTIBITList(i).name;
    soggPath = fullfile(folderPathEMOTIBIT, sogg);

    % File CSV nella sottocartella
    fileEMOTIBIT = dir(fullfile(soggPath, '*.csv')); % File CSV nella sottocartella

    % Messaggio errore per EMOTIBIT assente
    if isempty(fileEMOTIBIT)
        fprintf('Nessun file EMOTIBIT nella cartella %s\n', soggPath);
        continue;
    end

    epochTS = dataTS_unix(:,i).Epoch;
```

```

        sceltaTS = dataTS_unix{i}.Scelta(2:13);

% Array dei tipi di dati e nomi di file
dataTypes = {'EDA', 'Motion', 'PPG', 'Temperature'};

% Iterare attraverso i tipi di dati
for indice = 1:length(dataTypes)
    % Nome del tipo di dato corrente
    dataType = dataTypes{indice};

    % Verifica che ci siano 4 file CSV
    if length(fileEMOTIBIT) ~= 4
        fprintf('Errore: la cartella %s non contiene esattamente 4 file
CSV.\n', soggPath);
        continue;
    end

    % Leggi il file corrispondente
    dataTable = readtable(fullfile(soggPath, fileEMOTIBIT(indice).name));

    % Converti il timestamp a 13 cifre
    epochData = floor(dataTable.ts * 1000);
    dataTable.ts = epochData;

    % conversione epoch in data
    dataTableFormattata = datetime(dataTable.ts / 1000, 'ConvertFrom',
'posixtime', 'Format', 'dd-MM-yyyy HH:mm:ss.SSS', 'TimeZone', 'Europe/Rome');

    % aggiungere la colonna 'data' nella seconda posizione
    dataTable.data = dataTableFormattata;
    dataTable = dataTable(:, [1, end, 2:end-1]);

    % Intervalli di silenzio iniziale
    k = 1;
    timestamp1 = epochTS(k);
    timestamp1meno30 = timestamp1 - 30 * 1000;
    intervalloSilenzio1Min = timestamp1meno30 - 20 * 1000;
    intervalloSilenzio1Max = timestamp1meno30 + 10 * 1000;

    % Inizializzo le etichette
    labels1 = strings(height(dataTable), 1);
    labels2 = strings(height(dataTable), 1);
    dataTable.Labels1 = labels1;
    dataTable.Labels2 = labels2;
    dataTable.Scelta = strings(height(dataTable), 1);

    % labels 1
    dataTable.Labels1(epochData < intervalloSilenzio1Min) = "Connessione
Dispositivo";

```

```

        dataTable.Labels1(epochData >= intervalloSilenzio1Min & epochData <=
intervalloSilenzio1Max) = "Silenzio 1";
        dataTable.Labels1(epochData >= intervalloSilenzio1Max & epochData <=
timestamp1) = "Scarto Pre-inizio Test";

% labels 2
dataTable.Labels2(epochData < intervalloSilenzio1Min) = "Connessione
Dispositivo";
dataTable.Labels2(epochData >= intervalloSilenzio1Min & epochData <=
intervalloSilenzio1Max) = "Silenzio 1";
dataTable.Labels2(epochData >= intervalloSilenzio1Max & epochData <=
timestamp1) = "Scarto Pre-inizio Test";

% Itera per ogni timestamp successivo
for k = 1:12
    currentTimestamp = epochTS(k);
    nextTimestamp = epochTS(k + 1);

    % Calcola i limiti degli intervalli
    intervalA1_start = currentTimestamp;
    intervalA1_end = currentTimestamp + 30 * 1000;
    intervalLag1_start = intervalA1_end;
    intervalLag1_end = intervalLag1_start + 2 * 1000;
    intervalA2_start = intervalLag1_end;
    intervalA2_end = intervalA2_start + 30 * 1000;
    intervalLag2_start = intervalA2_end;
    intervalLag2_end = intervalLag2_start + 2 * 1000;
    intervalChoice_start = intervalLag2_end;
    intervalChoice_end = nextTimestamp;

    % Determina la combinazione corrente
    combinazioneCorrente = combinazioni(k);

    % Applica le etichette LABELS1 ai dati EMOTIBIT
    dataTable.Labels1(epochData >= intervalA1_start & epochData <
intervalChoice_end) = combinazioneCorrente;

    % Applica le etichette LABELS2 ai dati EMOTIBIT
    dataTable.Labels2(epochData >= intervalA1_start & epochData <
intervalA1_end) = extractBetween(combinazioneCorrente, 1, 2);
    dataTable.Labels2(epochData >= intervalLag1_start & epochData <
intervalLag1_end) = "Lag 2 Secondi";
    dataTable.Labels2(epochData >= intervalA2_start & epochData <
intervalA2_end) = extractBetween(combinazioneCorrente, 3, 4);
    dataTable.Labels2(epochData >= intervalLag2_start & epochData <
intervalLag2_end) = "Lag 2 Secondi";
    dataTable.Labels2(epochData >= intervalChoice_start & epochData <
intervalChoice_end) = "Scelta tra " + combinazioneCorrente;

```

```

        opzione1 = extractBetween(combinazioneCorrente, 1, 2); % estrae primo
suono
        opzione2 = extractBetween(combinazioneCorrente, 3, 4); % estrae secondo
suono

        % Determinare la scelta associata
        sceltaCorrente = sceltaTS(k);
        if sceltaCorrente == 1
            sceltaAssegnata = opzione1; % Scelta 1 corrisponde a opzione1
        elseif sceltaCorrente == 2
            sceltaAssegnata = opzione2; % Scelta 2 corrisponde a opzione2
        else
            sceltaAssegnata = "Errore"; % Caso di errore
        end
        dataTable.Scelta(epochData >= intervalA1_start & epochData <
intervalChoice_end) = sceltaAssegnata;
    end

    k=13;
    currentTimestamp = epochTS(k);
    nextTimestamp = epochTS(k + 1);

    % Applico le etichette LABELS 1 ai dati EMOTIBIT
    dataTable.Labels1(epochData >= currentTimestamp & epochData <=
nextTimestamp) = "Ancora qualche minuto...";

    % Applico le etichette LABELS 2 ai dati EMOTIBIT
    dataTable.Labels2(epochData >= currentTimestamp & epochData <=
nextTimestamp) = "Ancora qualche minuto...";

    k=14;
    currentTimestamp = epochTS(k);
    nextTimestamp = epochTS(k + 1);

    ts_fine = dataTable(end,1);
    ts_fine1 = table2array(ts_fine);

    % Calcolare i limiti degli intervalli
    silenzio2_start = currentTimestamp;
    silenzio2_end = silenzio2_start + 60 * 1000;
    ruscello_start = silenzio2_end;
    ruscello_end = ruscello_start + 60 * 1000;
    fastidioso1_start = ruscello_end;
    fastidioso1_end = fastidioso1_start + 60 * 1000;
    silenzio3_start = fastidioso1_end;
    silenzio3_end = silenzio3_start + 10 * 1000;
    fastidioso2_start = silenzio3_end;
    fastidioso2_end = fastidioso2_start + 60 * 1000;
    ruscello2_start = fastidioso2_end;
    ruscello2_end = ruscello2_start + 60 * 1000;

```

```

        silenzio4_start = ruscello2_end;
        silenzio4_end = nextTimestamp;

        % Applicare le etichette LABELS 1 ai dati nell'array per EMOTIBIT
        dataTable.Labels1(epochData >= currentTimestamp & epochData <=
nextTimestamp) = "Sequenza Finale";
        dataTable.Labels1(epochData >= nextTimestamp & epochData <= ts_fine1) =
"Disconnessione Dispositivi";

        % Applico le etichette LABELS 2 ai dati nell'array per EMOTIBIT
        dataTable.Labels2(epochData >= silenzio2_start & epochData < silenzio2_end)
= "Silenzio 2";
        dataTable.Labels2(epochData >= ruscello_start & epochData < ruscello_end) =
"Ruscello";
        dataTable.Labels2(epochData >= fastidioso1_start & epochData <
fastidioso1_end) = "Fastidioso";
        dataTable.Labels2(epochData >= silenzio3_start & epochData < silenzio3_end)
= "Silenzio 3";
        dataTable.Labels2(epochData >= fastidioso2_start & epochData <
fastidioso2_end) = "Fastidioso";
        dataTable.Labels2(epochData >= ruscello2_start & epochData < ruscello2_end)
= "Ruscello";
        dataTable.Labels2(epochData >= silenzio4_start & epochData < silenzio4_end)
= "Silenzio 4";
        dataTable.Labels2(epochData >= nextTimestamp & epochData <= ts_fine1) =
"Disconnessione Dispositivi";

        % Salva il risultato nella variabile corrispondente
        eval([dataType ' = dataTable;']);
    end

    DATA_EDA.(sogg) = EDA;
    DATA_MOTION.(sogg) = Motion;
    DATA_PPG.(sogg) = PPG;
    DATA_TEMPERATURE.(sogg) = Temperature;
end

```

Rimozione file Emotibit non validi

```

for i = 1:length(fileEMOTIBITList)
    sogg = fileEMOTIBITList(i).name;

    % Controllo dimensioni per ogni tipo di dato EDA
    if isfield(DATA_EDA, sogg) && ~isempty(DATA_EDA.(sogg)) % isfield per
controllare se il campo sogg esiste nella struct, isempty per verificare che il
campo sogg non sia vuoto
        fileEDA = dir(fullfile(folderPathEMOTIBIT, sogg, '*EDA*.csv'));
        if ~isempty(fileEDA)
            fileSizeKB = fileEDA.bytes / 1024;
            if fileSizeKB <= 300

```

```

        fprintf('ERRORE: File EDA per %s ha dimensione %.2f KB - Non supera
500 KB\n', sogg, fileSizeKB);
        DATA_EDA.(sogg) = []; % cella vuota se il file non supera la
dimensione minima
    end
else
    fprintf('ERRORE: Nessun file EDA trovato per %s\n', sogg);
    DATA_EDA.(sogg) = []; % Se non esiste il file, cella vuoto
end
end

% Controllo dimensioni per ogni dato PPG
if isfield(DATA_PPG, sogg) && ~isempty(DATA_PPG.(sogg))
    filePPG = dir(fullfile(folderPathEMOTIBIT, sogg, '*PPG*.csv'));
    if ~isempty(filePPG)
        fileSizeKB = filePPG.bytes / 1024;
        if fileSizeKB <= 800
            fprintf('ERRORE: File PPG per %s ha dimensione %.2f KB - Non supera
1000 KB\n', sogg, fileSizeKB);
            DATA_PPG.(sogg) = [];
        end
    else
        fprintf('ERRORE: Nessun file PPG trovato per %s\n', sogg);
        DATA_PPG.(sogg) = [];
    end
end

% Controllo dimensioni per ogni tipo di dato Motion
if isfield(DATA_MOTION, sogg) && ~isempty(DATA_MOTION.(sogg))
    fileMOTION = dir(fullfile(folderPathEMOTIBIT, sogg, '*Motion*.csv'));
    if ~isempty(fileMOTION)
        fileSizeKB = fileMOTION.bytes / 1024;
        if fileSizeKB <= 1200
            fprintf('ERRORE: File Motion per %s ha dimensione %.2f KB - Non
supera 1500 KB\n', sogg, fileSizeKB);
            DATA_MOTION.(sogg) = [];
        end
    else
        fprintf('ERRORE: Nessun file Motion trovato per %s\n', sogg);
        DATA_MOTION.(sogg) = [];
    end
end

% Controllo dimensioni per ogni dato Temperature
if isfield(DATA_TEMPERATURE, sogg) && ~isempty(DATA_TEMPERATURE.(sogg))
    fileTEMPERATURE = dir(fullfile(folderPathEMOTIBIT, sogg,
'*Temperature*.csv'));
    if ~isempty(fileTEMPERATURE)
        fileSizeKB = fileTEMPERATURE.bytes / 1024;
        if fileSizeKB <= 200

```

```
        fprintf('ERRORE: File Temperature per %s ha dimensione %.2f KB -  
Non supera 280 KB\n', sogg, fileSizeKB);  
        DATA_TEMPERATURE.(sogg) = [];  
    end  
else  
    fprintf('ERRORE: Nessun file Temperature trovato per %s\n', sogg);  
    DATA_TEMPERATURE.(sogg) = [];  
end  
end  
end
```

Appendice B

```
clear all
clc
close all

dataFolder = 'C:\Users\nucci\Desktop\tirocinio\codice\muse_data';
cd(dataFolder);

% Parametri
fs = 256;
dt = 1/fs;
df = 0.5;
nfft = round(fs / df);
noverlap = floor(nfft/2);
window = hanning(nfft);

% Definizione del filtro passa-banda
bpfilt = designfilt('bandpassiir', ...
    'FilterOrder', 4, 'HalfPowerFrequency1', 0.5, ...
    'HalfPowerFrequency2', 42, 'SampleRate', fs);

channels = {'TP9', 'AF7', 'AF8', 'TP10'};
colors_channels = [0 0.4470 0.7410;
    0.8500 0.3250 0.0980;
    0.9290 0.6940 0.1250;
    0.4940 0.1840 0.5560];

conditions = {'Pleasant', 'Annoying', 'Silence'};
colors_conditions = [0 0.4470 0.7410;
    0.8500 0.3250 0.0980;
    0.4660 0.6740 0.1880];

% Definizione dei soggetti da analizzare
subjects = 1:9;

% Loop per processare i dati di ogni soggetto
for jj = subjects
    filename = fullfile(dataFolder, strcat('S0', num2str(jj), '_data.csv'));

    app = readtable(filename, 'TextType', 'string');

    % **Estrazione indici delle condizioni**
    ind_pleasant = find(strcmp(app.Labels2, 'Ruscello'));
    ind_annoying = find(strcmp(app.Labels2, 'Fastidioso'));
    ind_silence = find(strcmp(app.Labels2, 'Silenzio 2'));

    % Seleziona solo il primo blocco di ogni condizione
    if ~isempty(ind_pleasant)
        end_pleasant = find(diff(ind_pleasant)~=1, 1, 'first');
        if isempty(end_pleasant)
            end_pleasant = length(ind_pleasant);
        end
    end
end
```

```

        end
        ind_pleasant = ind_pleasant(1):ind_pleasant(end_pleasant);
    end

    if ~isempty(ind_annoying)
        end_annoying = find(diff(ind_annoying)~=1, 1, 'first');
        if isempty(end_annoying)
            end_annoying = length(ind_annoying);
        end
        ind_annoying = ind_annoying(1):ind_annoying(end_annoying);
    end

    % Estrazione dati EEG
    tab_pleasant = app{ind_pleasant, 4:7};
    tab_annoying = app{ind_annoying, 4:7};
    tab_silence = app{ind_silence, 4:7};

    % Filtraggio
    filt_pleasant = detrend(filtfilt(bpfilt, tab_pleasant));
    filt_annoying = detrend(filtfilt(bpfilt, tab_annoying));
    filt_silence = detrend(filtfilt(bpfilt, tab_silence));

    % EEG Grezzi
    fig1 = figure;
    sgtitle(['Raw EEG - Soggetto ', num2str(jj)]);

    conditions_data = {tab_pleasant, tab_annoying, tab_silence};
    for c = 1:3
        subplot(3,1,c);
        hold on;
        for k = 1:4
            plot(0:dt:(size(conditions_data{c},1)-1)*dt, conditions_data{c}(:,k),
                'Color', colors_channels(k,:), 'LineWidth', 1.2);
        end
        hold off;
        ylabel([conditions{c}, ' (mV)']);
        xlabel('Time (s)');
        xlim ([0 60]);
        legend(channels, 'Location', 'eastoutside');
        grid on;
    end

    % EEG Filtrati
    fig2 = figure;
    sgtitle(['Filtered EEG - Soggetto ', num2str(jj)]);

    conditions_filt = {filt_pleasant, filt_annoying, filt_silence};
    for c = 1:3
        subplot(3,1,c);
        hold on;

```

```

        for k = 1:4
            plot(0:dt:(size(conditions_filt{c},1)-1)*dt, conditions_filt{c}(:,k),
'Color', colors_channels(k,:), 'LineWidth', 1.2);
            end
            hold off;
            ylabel([conditions{c}, ' (mV)']);
            xlabel('Time (s)');
            xlim ([0 60])
            legend(channels, 'Location', 'eastoutside');
            grid on;
        end

% PSD Grezzi
fig3 = figure;
sgtitle(['Raw PSD - Soggetto ', num2str(jj)]);

[pxx_pleasant, f] = pwelch(tab_pleasant, window, noverlap, nfft, fs);
[pxx_annoying, ~] = pwelch(tab_annoying, window, noverlap, nfft, fs);
[pxx_silence, ~] = pwelch(tab_silence, window, noverlap, nfft, fs);
psd_data = {pxx_pleasant, pxx_annoying, pxx_silence};

for k = 1:4
    subplot(2,2,k);
    hold on;
    for c = 1:3
        semilogy(f, psd_data{c}(:,k), 'Color', colors_conditions(c,:),
'LineWidth', 1.5);
    end
    hold off;
    set(gca, 'YScale', 'log');
    xlabel('Frequency (Hz)');
    ylabel('PSD (mV^2/Hz)');
    xlim([0 45])
    title([channels{k}]);
    legend(conditions, 'Location', 'eastoutside');
    grid on;
end

% PSD Filtrati
fig4 = figure;
sgtitle(['Filtered PSD - Soggetto ', num2str(jj)]);

[pxx_pleasant_filt, ~] = pwelch(filt_pleasant, window, noverlap, nfft, fs);
[pxx_annoying_filt, ~] = pwelch(filt_annoying, window, noverlap, nfft, fs);
[pxx_silence_filt, ~] = pwelch(filt_silence, window, noverlap, nfft, fs);
psd_data_filt = {pxx_pleasant_filt, pxx_annoying_filt, pxx_silence_filt};

for k = 1:4
    subplot(2,2,k);
    hold on;

```

```

        for c = 1:3
            semilogy(f, psd_data_filt{c}(:,k), 'Color', colors_conditions(c,:),
'LineWidth', 1.5);
        end
        hold off;
        set(gca, 'YScale', 'log');
        xlabel('Frequency (Hz)');
        ylabel('PSD (mV^2/Hz)');
        xlim([0 45])
        title([channels{k}]);
        legend(conditions, 'Location', 'eastoutside');
        grid on;
    end
end

```

Bibliografia

- [1] Genuit, K., & Fiebig, A. (2005). Prediction of psychoacoustic parameters. Journal of the Acoustical Society of America.
- [2] Sanei, S. and Chambers, J.A. (2007) EEG Signal Processing. John Wiley & Sons Ltd., 35-125.
- [3] Muse Research. Muse Official Website. Disponibile su: <https://choosemuse.com/pages/muse-research>
- [4] Muse S Gen 2. Muse Official Website. Disponibile su: <https://eu.choosemuse.com/products/muse-s-gen-2-6-pack>
- [5] Chan, K.S.N., Srisurangkul, C., Depaiwa, N., & Pangkreung, S. (2021). Detection of Driver Drowsiness from EEG Signals Using Wearable Brain Sensing Headband. Journal of Research and Applications in Mechanical Engineering, 9(1), 116-126.
- [6] Kim, J., Kim, M., Jang, M., & Lee, J. (2022). The Effect of Juingong Meditation on the Theta to Alpha Ratio in the Temporoparietal and Anterior Frontal EEG Recordings. International Journal of Environmental Research and Public Health, 19(3), 1721.
- [7] Ratti, E., Waninger, S., Berka, C., Ruffini, G., & Verma, A. (2017). Comparison of Medical and Consumer Wireless EEG Systems for Use in Clinical Trials.
- [8] Krigolson, O., Williams, C., Norton, A., & Colino, F. (2017). Choosing MUSE: Validation of a Low-Cost, Portable EEG System for ERP Research.

- [9] Luque, J. R., Maron, M. J., & Casilari, E. (2012). Analytical and empirical evaluation of the impact of Gaussian noise on the modulations employed by Bluetooth Enhanced Data Rates.
- [10] Horvers, A., Tombeng, N., Bosse, T., Lazonder, A. W., & Molenaar, I. Detecting Emotions through Electrodermal Activity in Learning Contexts: A Systematic Review.