
ALMA MATER STUDIORUM – UNIVERSITÀ DI BOLOGNA

CAMPUS DI CESENA

SCUOLA DI INGEGNERIA E ARCHITETTURA

CORSO DI LAUREA IN INGEGNERIA BIOMEDICA

**SVILUPPO DI UN' INTERFACCIA GRAFICA UTENTE IN MATLAB PER
LA VISUALIZZAZIONE E L'ANALISI SPETTRALE DI SEGNALI
ELETTOENCEFALOGRAFICI ED ELETTROCARDIOGRAFICI**

Elaborato in

STRUMENTAZIONE BIOMEDICA

Relatore

Prof.ssa ELISA MAGOSSO

Presentato da

GIANLUCA CARLONI

ANNO ACCADEMICO 2017/2018

Sommario

Introduzione	5
---------------------------	----------

Capitolo 1

Basi anatomiche e fisiologiche dei potenziali EEG ed ECG e metodi di analisi	9
1.1. Elettroencefalografia	10
1.2. Cenni di anatomia dell'encefalo: telencefalo e corteccia cerebrale	12
1.3. Basi neurofisiologiche dei segnali EEG di superficie	14
1.4. Analisi EEG nel dominio delle frequenze e ritmi EEG	19
1.5. Elettrocardiografia.....	25
1.6. Basi cardiofisiologiche del segnale ECG	26
1.7. Heart Rate (HR) e Heart Rate Variability (HRV).....	28
1.8. Importanza della HRV e suoi metodi di analisi	30
1.9. Bande di potenza LF ed HF dell'HRV.....	31

Capitolo 2

Descrizione delle esperienze in laboratorio per l'acquisizione dei segnali EEG ed ECG.....	34
2.1. Il Laboratorio di Realtà Virtuale e l'ambiente "CAVE"	34
2.2. Finalità dell'esperimento nel "CAVE"	36
2.3. Fasi dell'esperimento nel "CAVE"	39
2.4. Setup strumentale utilizzato nell'esperimento nel "CAVE": il sistema di acquisizione "OpenBCI"	43
2.5. Il Laboratorio di Ingegneria Biomedica (LIB).....	45
2.6. Finalità degli esperimenti in LIB	46
2.7. Fasi degli esperimenti in LIB	46
2.8. Setup strumentale utilizzato negli esperimenti in LIB: il sistema di acquisizione "Neurowave"	50
2.9. Considerazioni aggiuntive	54

Capitolo 3

Interfacce Grafiche Utente: introduzione a “GUIDE” e oggetti in MATLAB	57
3.1. Le interfacce grafiche utente in MATLAB	57
3.2. L’ambiente GUIDE di MATLAB e il Layout Editor di GUIDE	58
3.3. Gli oggetti.....	60
3.4. Le proprietà degli oggetti	63
3.5. Generazione del codice	69

Capitolo 4

Organizzazione e funzionalità dell’Interfaccia Grafica Utente realizzata	74
4.1. Struttura e suddivisione della GUI.....	76
4.2. Scelta sistema di acquisizione e caricamento dati.....	78
4.3. Inserimento della frequenza di campionamento.....	80
4.4. Visualizzazione EEG nel dominio del tempo	80
4.5. Operazioni di filtraggio in frequenza	84
4.6. Stima della densità spettrale di potenza del segnale	85
4.7. Calcolo della potenza del segnale in specifiche bande	88
4.8. Visualizzazione del segnale ECG e analisi dell’HRV	92

Capitolo 5

Codice MATLAB di implementazione dell’Interfaccia Grafica Utente.....	97
5.1. Selezione del sistema di acquisizione	97
5.2. Creazione del pop-up menu degli elettrodi e visualizzazione Single-channel	100
5.3. Eliminazione del valor medio e filtraggio Notch.....	101
5.4. Generazione del Multi-channel Plot per visualizzazione simultanea degli elettrodi.....	103
5.5. Filtraggi passa-basso e passa-alto	104
5.6. Calcolo della densità spettrale di potenza per le fasi dell’esperimento.....	106
5.7. Calcolo della potenza in una specifica banda dei segnali EEG e salvataggio dei risultati	112
5.8. Visualizzazione del segnale ECG ed analisi dell’HRV	116

Conclusioni	119
Indice delle Figure e delle Tabelle	124
Bibliografia e fonti	126

Parole chiave:

Matlab

Graphical User Interface

Densità spettrale di potenza

Potenza in banda alpha e beta

Heart Rate Variability

INTRODUZIONE

Il segnale elettroencefalografico (EEG) è la registrazione di potenziali elettrici che si generano sulla superficie dello scalpo in seguito all'attività elettrica delle cellule nervose della corteccia cerebrale. Il segnale elettrocardiografico (ECG) è la registrazione dei potenziali generati sulla superficie del corpo in seguito al processo di stimolazione della muscolatura cardiaca. La misura di questi segnali è di larghissimo impiego sia nella pratica clinica che nella ricerca per la facilità di acquisizione, la non invasività, il costo relativamente basso della strumentazione necessaria alla loro misura. Lo studio delle caratteristiche nel tempo e del contenuto frequenziale dei segnali EEG ed ECG è importante non solo per avere informazioni su eventuali disordini o patologie del sistema nervoso centrale (EEG), cardiaco ed autonomico (ECG), ma anche per migliorare la comprensione della fisiologia di questi sistemi. Da questo punto di vista, di grande interesse per la ricerca è lo studio delle modificazioni di questi segnali in soggetti sani in risposta a specifiche stimolazioni esterne, all'esecuzione di task e a conseguenti stati emotivi e attentivi, rispetto a condizioni di riposo (in assenza di stimolazioni). A tale scopo, il primo passo è quello di ottenere i suddetti segnali. Avendo a disposizione un laboratorio debitamente attrezzato ed un sistema di acquisizione appropriato, è possibile eseguire alcuni esperimenti *ad hoc* con soggetti volontari sani ed estrarre da questi i segnali EEG ed ECG. In questo contesto, grazie alla disponibilità di laboratori con adeguate strumentazioni e tecnologie, recentemente un gruppo di docenti dei Dipartimenti DEI (in particolare del raggruppamento di Bioingegneria) e DIN dell'Università di Bologna ha eseguito due esperimenti su soggetti volontari sani, registrandone l'EEG ed ECG. Il primo di questi esperimenti aveva come obiettivo lo studio di tali segnali durante diversi compiti attentivi svolti dal soggetto; il secondo aveva come obiettivo lo studio di tali segnali quando il soggetto veniva immerso in ambienti di realtà virtuale. Il secondo fondamentale passo è l'analisi di tali segnali, attraverso una loro opportuna elaborazione: in questo ambito è di grande utilità avere a disposizione un sistema software che permetta di caricare questi dati, eseguire le elaborazioni su di essi, e generare risultati, sulla base dei quali trarre conclusioni significative per l'esperimento realizzato. È da notare che sebbene molti sistemi di acquisizione dei segnali EEG ed ECG dispongano già di

pacchetti software per l'elaborazione dei segnali acquisiti, tuttavia, tali software hanno spesso costi molto elevati, e in ogni caso non permettono analisi ottimizzate per lo specifico scopo e per lo specifico esperimento eseguito. In altri casi, invece, come nel caso di sistemi di acquisizione più economici, tali software sono del tutto assenti. È proprio in questo contesto che si è svolto il mio lavoro di tesi. Tale lavoro ha riguardato la realizzazione in MATLAB® di un programma, più precisamente una Interfaccia Grafica Utente (Graphical User Interface), che consenta ad un utilizzatore di visualizzare in modo appropriato i segnali acquisiti nei due esperimenti citati e quindi di condurre un'opportuna analisi di tali segnali. Nello specifico, ci si è concentrati sul mettere a disposizione, attraverso l'Interfaccia Grafica Utente, strumenti per l'analisi spettrale dei segnali EEG ed ECG. Più precisamente, sia per il segnale EEG sia per il segnale di frequenza cardiaca (ottenuto dal segnale ECG) ci si è concentrati sull'estrazione della densità spettrale di potenza e sul calcolo della potenza in specifiche bande. Infatti, variazioni di potenza tra le varie bande di questi segnali possono fornire indicazioni sullo stato di attivazione del sistema nervoso centrale ed autonomico legate alle stimolazioni e sollecitazioni esterne. Mettendo a disposizione tali strumenti è dunque possibile investigare la presenza di certe correlazioni tra le caratteristiche spettrali dei segnali e lo stato attentivo ed emotivo del soggetto modulato durante l'esperimento.

L'elaborato è organizzato nel seguente modo.

Il Capitolo 1 è dedicato alle basi anatomiche e fisiologiche dei potenziali EEG ed ECG, ed ai metodi di analisi di tali biopotenziali. Esso inizia con una breve panoramica sulla storia dell'elettroencefalografia; segue una trattazione sull'anatomia del telencefalo e della corteccia cerebrale; prosegue descrivendo i principi neurofisiologici dei segnali EEG di superficie. In seguito, il capitolo introduce il lettore alle principali analisi EEG nel dominio delle frequenze e ai cosiddetti ritmi EEG (ovvero fluttuazioni del segnale EEG in specifiche bande di frequenza). Nella seconda metà del capitolo, prima sono dati cenni sulla storia dell'elettrocardiografia e sulla fisiologia su cui si basa; poi si spiega come venga estratta la frequenza cardiaca del segnale ECG e l'importanza della variabilità della frequenza cardiaca (Heart Rate Variability, ovvero le variazioni continue nel valore di frequenza cardiaca); infine, si illustrano i principali metodi di analisi del segnale ECG nel dominio dei tempi e, soprattutto, in quello delle frequenze.

Il Capitolo 2 descrive le due esperienze in laboratorio (a cui ho in parte partecipato insieme ad altri due tesisti, uno triennale e uno magistrale), volte ad acquisire i segnali EEG ed ECG. Per ciascuno dei due esperimenti, viene descritto il laboratorio in cui è stato svolto, il setup strumentale, il sistema d'acquisizione utilizzato, l'obiettivo di studio di ciascun esperimento e vengono dettagliate le fasi di svolgimento dell'esperimento.

Il Capitolo 3 fornisce una panoramica sulle interfacce grafiche utente in MATLAB e sui metodi per la loro creazione (programmatica o interattiva). Contiene informazioni relative a "GUIDE" (Graphical User Interface Design Environment), l'ambiente grafico utilizzato per la realizzazione del programma; mostra i principali oggetti disponibili, descrive il funzionamento e gli strumenti utilizzati per lo sviluppo delle interfacce. Il capitolo è arricchito da due esempi di interfacce grafiche, realizzate *ad hoc* per aiutare il lettore a familiarizzare con terminologie e funzionalità.

Nel Capitolo 4 vengono presentate la struttura e la funzionalità dell'interfaccia grafica che ho realizzato, come una sorta di manuale utente. In particolare, dapprima ne viene descritto l'aspetto grafico, in seguito si spiega come utilizzare le principali funzionalità messe a disposizione dall'interfaccia e si illustra il risultato grafico e/o numerico restituito all'utente quando vengono eseguite tali funzionalità. Ad esempio: tra le varie funzionalità messe a disposizione vengono illustrate come l'utente possa selezionare il sistema di acquisizione utilizzato nell'esperimento (sistemi diversi comportano un diverso numero di segnali e di frequenza di campionamento), come possano essere visualizzati i segnali EEG registrati, sia nella loro forma grezza, sia dopo opportune operazioni di filtraggio, come si possa procedere al calcolo della potenza del segnale EEG, come si possano visualizzare i segnali ECG, e come si possa svolgere l'analisi in frequenza dell'HRV.

Per concludere, il Capitolo 5 descrive il codice Matlab di implementazione dell'interfaccia grafica, e pertanto è caratterizzato da una descrizione non più dal punto di vista dell'utente (livello front-end) ma da quello del programmatore (livello del codice). Infatti, per ciascuna delle funzionalità che l'interfaccia grafica mette a disposizione dell'utente viene riportato e spiegato il codice Matlab

corrispondente, permettendo al lettore interessato di comprendere come avviene l'implementazione effettiva delle varie funzionalità dell'interfaccia.

CAPITOLO 1

BASI ANATOMICHE E FISIOLOGICHE DEI POTENZIALI EEG ED ECG E METODI DI ANALISI

Il contesto operativo in cui si inserisce questo elaborato è quello dell'acquisizione ed elaborazione di segnali fisiologici per lo studio delle loro modificazioni quando un soggetto è sottoposto a specifiche stimolazioni o impegnato in specifici task. Questa tesi si colloca infatti all'interno di un vasto progetto (portato avanti da un gruppo di docenti di Bioingegneria dell'Università di Bologna) che si occupa di investigare quali modificazioni nei parametri fisiologici possono riflettere maggiormente lo stato emotivo e attentivo di un soggetto. A tale scopo, come verrà descritto in un successivo capitolo, i ricercatori coinvolti nel progetto hanno ideato due specifici esperimenti: un esperimento è finalizzato a valutare l'effetto su un soggetto di diverse componenti attentive (visiva, motoria, mentale); un altro esperimento è finalizzato a investigare l'effetto di ambienti di realtà virtuale dal punto di vista dello stato di eccitazione/gradimento che un ambiente di realtà virtuale può elicitarne in un soggetto.

Nello specifico, è stata posta l'attenzione su due potenziali bioelettrici: l'*elettroencefalogramma (EEG)* e l'*elettrocardiogramma (ECG)*. La scelta è ricaduta su questi per la loro natura spontanea, per la loro facilità di acquisizione e quindi non invasività, e soprattutto per la loro suscettibilità in risposta a modificazioni dello stato attentivo, di stress, di gradimento e di coinvolgimento del soggetto. Come è noto, tali potenziali sono il risultato dell'attività elettrochimica di una classe di cellule dette *cellule eccitabili*. Tali potenziali sono stati acquisiti sui soggetti coinvolti nei due esperimenti.

Gli esperimenti sono stati svolti presso il Laboratorio di Ingegneria Biomedica (LIB) dell'Università di Bologna - Campus di Cesena e nel Laboratorio di Realtà Virtuale dell'Università di Bologna – Campus di Forlì, e i segnali EEG ed ECG acquisiti mediante i sistemi di acquisizione *Neurowave* e *OpenBCI*.

All'interno di questo vasto progetto, l'obiettivo specifico del presente lavoro di tesi è stato quello di realizzare una interfaccia grafica *GUI (Graphical User Interface)*

in MATLAB® al fine di permettere una appropriata visualizzazione dei segnali acquisiti negli esperimenti e quindi di condurre un'analisi spettrale dei segnali, così da permettere di investigare la presenza di correlazioni tra le caratteristiche spettrali dei segnali e lo stato emotivo/attentivo del soggetto modulato durante l'esperimento. In particolare, poiché ciascuno dei due esperimenti ideati prevedeva fasi di assenza di compito e di stimolazione alternate a fasi di stimolazione ed esecuzione di compiti, la GUI si è concentrata sulla computazione e visualizzazione delle caratteristiche spettrali in queste diverse fasi per evidenziarne, graficamente e quantitativamente, eventuali differenze.

1.1 ELETTROENCEFALOGRAFIA

Per secoli gli studi sull'encefalo si sono limitati alla mera descrizione anatomica. Ancora a metà dell'800, vi era un mondo sconosciuto: la fisiologia elettrica cerebrale. Sotto la scia degli studi di Galvani e Volta riguardo all'elettrofisiologia, il primo a riuscire a registrare l'attività elettrica nel cervello esposto di un animale fu l'inglese Richard Caton nel 1875. Fu però solo nel 1924 (e poi nel 1930) che il medico tedesco Hans Berger effettuò la registrazione dell'attività elettrica cerebrale sulla superficie del cranio di soggetti umani. Fu lui a coniare il termine *Elektenkephalogram* e a rilevare la presenza delle onde alpha (poi dette onde Berger) e delle onde beta. L'EEG divenne progressivamente una pratica di routine in tutto il mondo occidentale, vedendo negli anni '40 e '50 il moltiplicarsi di aziende produttrici americane, europee e giapponesi, negli anni '80 l'emergere degli elettroencefalografi digitali, nel 1994 l'affermarsi di standard di registrazione di EEG di superficie, e negli anni 2000 l'utilizzo dell'EEG in nuovi ambiti come *Unità di Terapia Intensiva (UTI)*, sale operatorie, ma soprattutto nelle Neuroscienze cognitive. Queste ultime infatti si prefiggono di descrivere come i processi cognitivi nascono dai processi cerebrali. [1]

Oggi, con il termine *elettroencefalografia* si designa la registrazione dell'attività elettrica cerebrale, quale metodica fondamentale per valutare lo stato funzionale del cervello. Tale attività può essere registrata convenzionalmente mediante tre tipi di elettrodi, e in base a ciò è possibile definire le tre seguenti famiglie: se gli elettrodi sono posizionati sullo scalpo, si parla di *elettroencefalografia di superficie* o

standard (EEG); con elettrodi posti direttamente sulla superficie della corteccia cerebrale, si parla di *elettrocorticografia* (ECoG); con elettrodi ad ago posti in profondità nel tessuto cerebrale, si parla di *stereolettroencefalografia* (SEEG). L'elettroencefalografia standard, data la sua non invasività, è la metodica più diffusamente utilizzata nelle registrazioni cliniche di routine, sia in pazienti ambulatoriali che ricoverati. [1]

Formalmente, l'elettroencefalogramma è una rappresentazione grafica (vedi Figura. 1.1) della differenza di potenziale fra due diverse posizioni in funzione del tempo. Il segnale EEG generato sullo scalpo dai neuroni cerebrali è modificato dalle proprietà di conduzione elettrica dei tessuti frapposti tra la sorgente elettrica e l'elettrodo di registrazione, dalle proprietà di conduzione dell'elettrodo stesso, e dall'orientazione del generatore corticale rispetto all'elettrodo.

È possibile ottenere l'EEG grazie al processo di *volume conduction*: un flusso di corrente attraversa i tessuti tra il generatore elettrico e l'elettrodo di registrazione. L'EEG fornisce una proiezione bi-dimensionale di una realtà tri-dimensionale, il che significa che teoricamente è impossibile determinare la localizzazione del generatore di EEG basandosi unicamente sulle informazioni EEG registrate su scalpo. Ci si riferisce a ciò come *problema inverso*. [2]

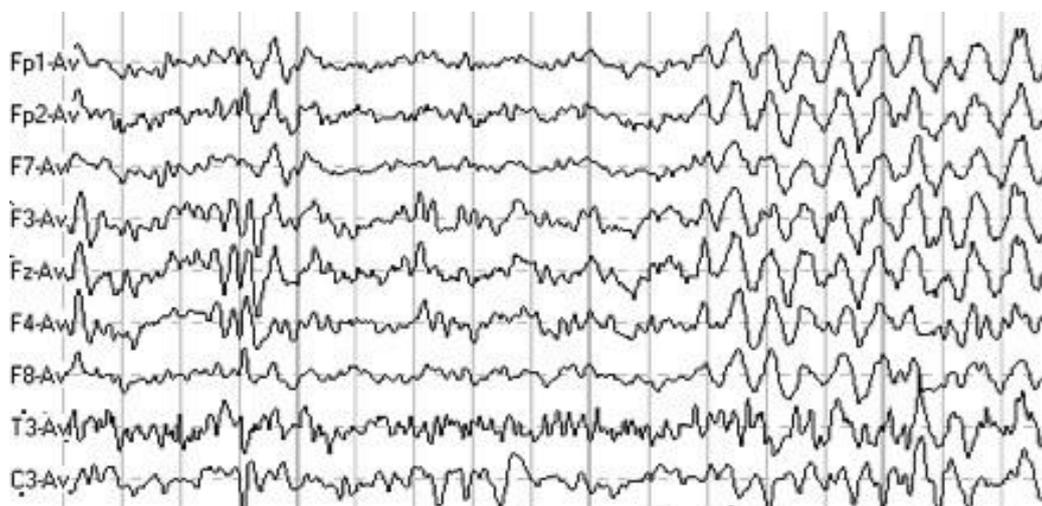


Figura. 1.1 Tipici tracciati di un elettroencefalogramma di superficie (tratta da [3])

1.2 CENNI DI ANATOMIA DELL'ENCEFALO: TELENCEFALO E CORTECCIA CEREBRALE

Più di duemila anni fa, Aristotele dichiarò che il cuore è la sede dell'anima. Oggi, al contrario, si ritiene che l'encefalo sia l'organo che determina le caratteristiche della specie umana e che la differenzia dalle altre, grazie alla possibilità di originare comportamenti complessi, come parlare, scrivere una sinfonia o creare un mondo immaginario nella mente. Si può studiare l'encefalo a molti livelli di organizzazione: da quelli che guardano ai singoli neuroni e a ciò che succede in essi in risposta a determinati segnali chimici ed elettrici, a quelli integrativi che guardano a gruppi di neuroni e a come essi reagiscono costituendo circuiti o reti nervose. Un principio base da tenere sempre presente quando si studia l'encefalo è che ogni funzione, anche semplice come flettere un dito, può coinvolgere diverse regioni dell'encefalo e anche del midollo spinale. [4]

Il *telencefalo* (o cervello) è la parte più voluminosa e più caratteristica dell'encefalo umano, e occupa la maggior parte della cavità cranica. È costituito da due emisferi connessi tra loro dal *corpo calloso*, una struttura costituita da assoni che passano da un lato all'altro dell'encefalo. Questa connessione permette ai due emisferi di comunicare e di cooperare fra loro. [4]

Le funzioni cognitive più complesse e di più alto livello risiedono nella *corteccia cerebrale*, la parte più esterna del telencefalo, spesso pochi millimetri (tipicamente < 5mm). La corteccia cerebrale ha una conformazione convoluta, ricca di giri, valli e scissure, che ricorda quella del gheriglio di una noce. Il grado di ripiegamento del cervello è direttamente collegato al livello di elaborazione di cui è capace: i mammiferi meno evoluti, per esempio i roditori, hanno una superficie cerebrale praticamente liscia. [4]

In ogni emisfero, la corteccia cerebrale è suddivisa in quattro *lobi* (vedi Figura 1.2): frontale, parietale, temporale, occipitale. La *corteccia visiva primaria* è situata nel lobo occipitale, e contiene una mappa topologica dei segnali della retina. La *corteccia uditiva primaria* è situata nel lobo temporale, e ha una organizzazione tonotopica, ovvero neuroni vicini tra loro codificano toni tra loro vicini. La

corteccia somatosensoriale primaria è situata nel lobo parietale, e riceve informazioni dai recettori generali di senso (pressione, tatto, dolore) della pelle. La *corteccia motoria primaria* è situata nel lobo frontale e contiene i neuroni che proiettano direttamente al midollo spinale o al tronco encefalico per attivare i cosiddetti motoneuroni per la contrazione muscolare. Tuttavia, molte aree della corteccia non sono né sensoriali né motorie, e sono dette *aree associative*; esse sono situate principalmente nei lobi frontali e parietali, con l'importantissima funzione di integrare e associare le informazioni provenienti da più modalità sensoriali. [1]

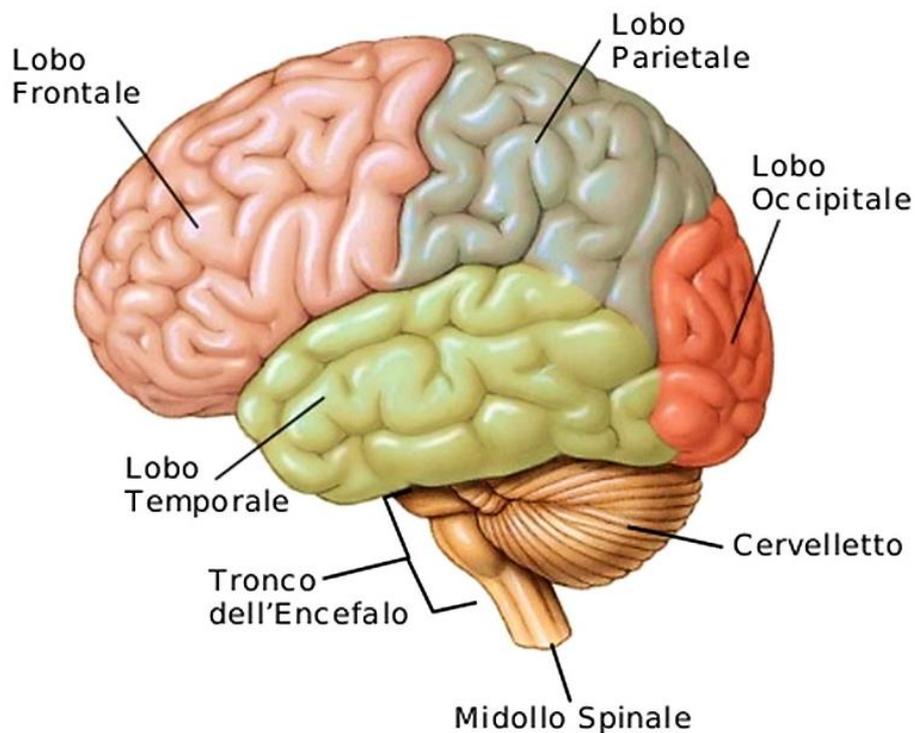


Figura 1.2. Vista laterale dell'encefalo: suddivisione dei 4 lobi principali (tratto da [5])

1.3 BASI NEUROFISIOLOGICHE DEI SEGNALI EEG DI SUPERFICIE

I segnali elettroencefalografici che si registrano sullo scalpo sono fenomeni di tipo macroscopico, riflesso dell'attività sincronizzata di enormi popolazioni neuronali; tali segnali sono detti *potenziali di campo*. Dal momento che la corteccia cerebrale è la struttura più prossima allo scalpo (dove i segnali vengono prelevati), l'EEG di superficie riflette prioritariamente l'attività dei neuroni corticali, mentre strutture sottocorticali quali talamo, ippocampo e tronco encefalico contribuiscono in modo minoritario ed indiretto. [1]

La corteccia cerebrale è organizzata in sei strati funzionali, detti *strati corticali*, disposti secondo la profondità (vedi Figura 1.3).

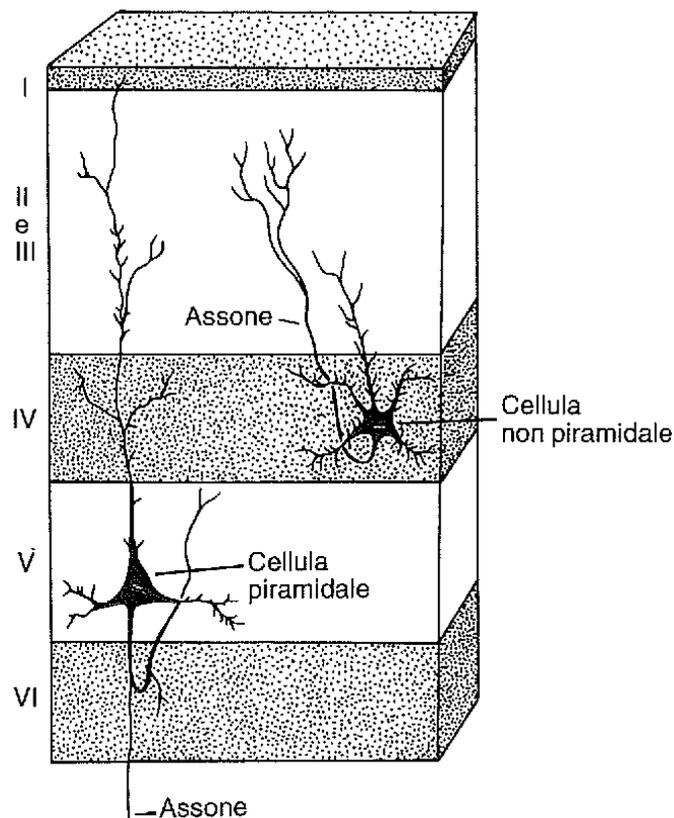


Figura.1.3. Suddivisione in strati della corteccia cerebrale (tratta da [6])

I neuroni nella corteccia sono di due tipi: piramidali e non piramidali. I neuroni piramidali, detti *neuroni di proiezione*, sono principalmente presenti negli strati III, V e VI, e sono caratterizzati da un grande soma ed un dendrite apicale molto lungo (è detto apicale in quanto si diparte dall'apice del soma).

I dendriti apicali dei neuroni piramidali sono tutti disposti parallelamente gli uni agli altri e perpendicolarmente alla superficie corticale. Al contrario, i neuroni non piramidali, detti *neuroni locali*, hanno corpi cellulari più piccoli e dendriti molto brevi, che si dipartono dal soma in tutte le direzioni. Data l'organizzazione coerente dei neuroni piramidali, non stupisce il fatto che siano proprio i potenziali postsinaptici di tali neuroni i principali generatori dell'EEG di superficie. [1]

Ogni sinapsi si comporta come una batteria trasportando corrente in un piccolo loop. Sia i potenziali post-sinaptici di eccitazione (EPSPs) sia di inibizione (IPSPs) contribuiscono all'attività sinaptica che dà origine al segnale EEG di superficie. Le correnti extracellulari dovute ai potenziali post-sinaptici si sviluppano abbastanza lentamente (fenomeni della durata delle decine di millisecondi) così da consentire una somma temporale e permettere la generazione di potenziali sufficientemente elevati da poter essere registrati in superficie.

La corrente che fluisce attraverso la resistenza esterna del tessuto corticale si somma con le correnti di loop dei neuroni vicini ed insieme costituiscono un campo medio locale. Viste dal di fuori delle cellule, le aree di membrana dove la corrente fluisce verso l'esterno o verso l'interno delle cellule sono chiamate rispettivamente *sources* and *sinks*. [2]

Le correnti di eccitazione (cioè associate a EPSPs), che coinvolgono ioni Na^+ e Ca^{2+} , fluiscono verso l'interno della membrana cellulare in corrispondenza della sinapsi eccitatoria e tendono a fluire verso l'esterno della membrana in corrispondenza a regioni cellulari lontano dalla sinapsi. La corrente che fluisce all'esterno della membrana ritorna verso il sito sinaptico attraverso il fluido esterno extracellulare. La corrente esterna è detta *corrente di ritorno passivo* (dallo spazio intracellulare a quello extracellulare). Le correnti di inibizione (cioè associate a IPSPs), che coinvolgono ioni Cl^- e K^+ , fluiscono nella direzione opposta. [2]

Per meglio comprendere la fisiologia che giace dietro la generazione del segnale EEG, occorre comprendere i flussi di corrente nel dendrite apicale di un neurone piramidale. Si dia la seguente situazione: sinapsi eccitatoria nella regione del dendrite più prossima alla superficie della corteccia (vedi Figura 1.4b).

In tale sito si avrà un flusso di cariche positive entranti che determina una elettronegatività nello spazio extracellulare. Le cariche positive percorrono il dendrite ed escono attraverso la membrana in corrispondenza dei siti basali, quindi più distanti, creando una elettropositività nel fluido extracellulare in quei siti. In tal modo, considerando ciò che avviene nel fluido extracellulare, si viene a creare una classica configurazione a dipolo, col polo negativo (sink) in alto e quello positivo (source) in basso.

Se, nell'ipotesi ideale che si possa registrare l'attività di un singolo neurone, posizionassimo un elettrodo in prossimità del neurone esso registrerebbe una elettronegatività extracellulare, che per convenzione è rappresentata sul tracciato come una deflessione verso l'alto. [1]

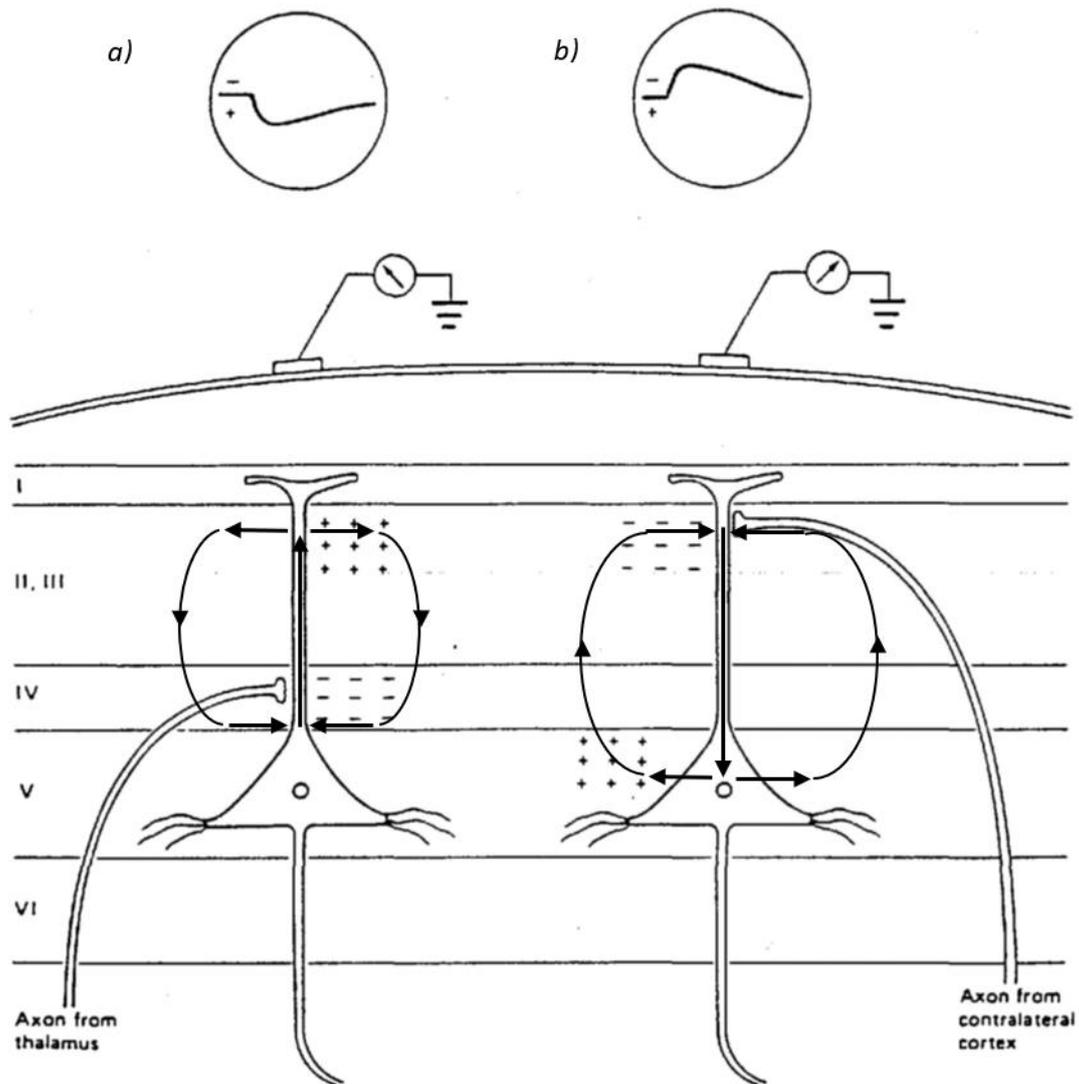


Figura 1.4. Esempi di sinapsi eccitatorie nelle regioni basali (a) e apicali (b) del dendrite di un neurone piramidale. La corrente in questo caso è dovuta a flussi di cariche positive. Nel primo caso l'elettrodo di superficie registra una elettropositività extracellulare, a cui associa una deflessione verso il basso. Nel secondo caso l'elettrodo registra una elettro negatività extracellulare, a cui associa una deflessione verso l'alto; (tratta da [2])

Al sito sinaptico di un EPSP vi è un *sink* di corrente attiva (campo extracellulare negativo). Ioni positivi migrano verso la cellula e ne depolarizzano la membrana. Nella parte distale della cellula (corpo e dendriti distali) un *source* di corrente passiva fuori dalla cellula è associato a campo extracellulare positivo. [2]

Le principali sorgenti di EEG derivano dalla corteccia cerebrale e formano campi di potenziale tridimensionali, che possono essere registrati come campi bidimensionali proiettati (e quindi potenziali elettrici) in funzione del tempo. Come precedentemente menzionato, i principali generatori di campi EEG misurati sulla superficie del cervello o sullo scalpo sono in potenziali sinaptici, sia eccitatori che inibitori dei neuroni piramidali.

Riassumendo, l'elettroencefalogramma nasce dall'attività sinaptica sincronizzata di popolazioni di neuroni. L'eccitazione dei neuroni postsinaptici crea un potenziale extracellulare vicino ai dendriti che è più negativo di qualsiasi altra zona lungo il neurone.

Questa situazione è un dipolo: una regione a carica positiva (*source*) ed una regione a carica negativa (*sink*) separate da una certa distanza. Gli elettrodi rilevano la somma delle cariche positive e negative in loro prossimità. Nel caso in cui un elettrodo è equidistante da *source* e *sink* di un dipolo, esso misurerà una neutralità netta; ciò significa che un elettrodo può rilevare dipoli solo quando è più vicino al terminale positivo o al terminale negativo del dipolo.

Si possono misurare due principali tipi di dipolo: *dipoli radiali* (Figura 1.5a), orientati perpendicolarmente alla superficie, e *dipoli tangenziali* (Figura 1.5b), orientati parallelamente alla superficie dello scalpo [7]. I dipoli hanno un lato positivo e uno negativo, pertanto produrranno una deflessione positiva o negativa in regioni diverse dello scalpo.

Dal momento che gli elettrodi rilevano la somma di cariche circostanti, i dipoli di più neuroni nella stessa regione si sommano insieme, fornendo una misura che riflette la situazione di un singolo dipolo di ampiezza dipendente dal numero di neuroni i cui dipoli hanno contribuito a sommarsi. [8]

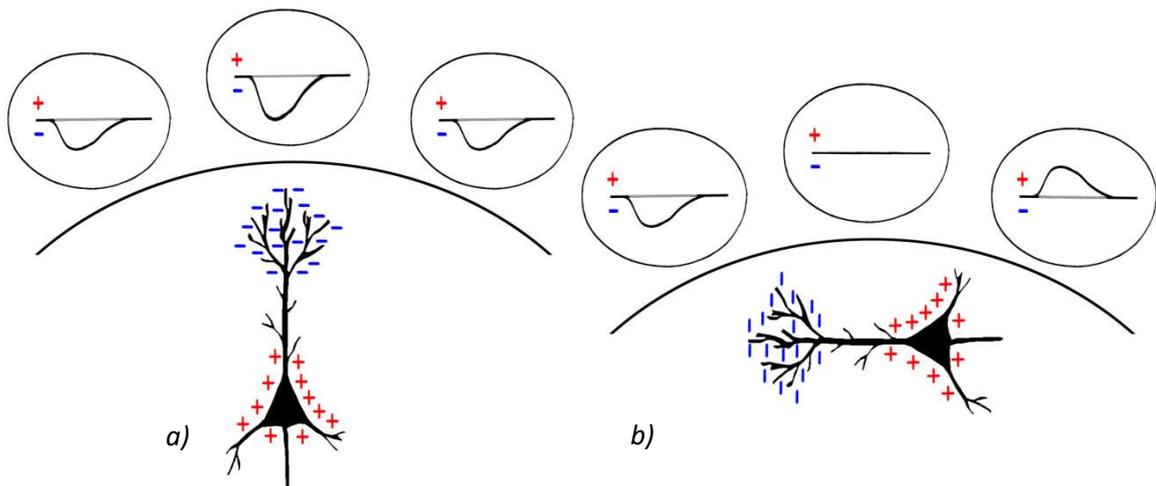


Figura 1.5. Misurazione dei dipoli radiali (a) e tangenziali (b). I dipoli radiali (a) producono deflessioni in una sola direzione, mentre i dipoli tangenziali (b) producono deflessioni in entrambe le direzioni. (tratta da [9])

1.4 ANALISI EEG NEL DOMINIO DELLE FREQUENZE E RITMI EEG

Come già detto, l'EEG registra la variazione nel tempo di potenziali elettrici a diversi siti sullo scalpo. Campionato ad una frequenza fino a 1000 Hz, con un numero di elettrodi fino a 2^8 , per un tempo fino a due ore continuative, l'EEG, per esempio, potrebbe generare una matrice di 7.2 milioni di campioni per 256 siti, per un totale di oltre 1.8 miliardi di dati. Esistono numerose tecniche di analisi per cercare di ottenere qualche conoscenza da questa massa di dati. Una delle più utili è *l'analisi spettrale di potenza (spectral power analysis)*, che ci permette di misurare quanto i neuroni che generano l'EEG stanno oscillando in modo sincrono alle varie frequenze. [10]

Il teorema di Fourier afferma che ogni serie periodica di oscillazioni può essere analizzata in pacchetti di possibili oscillazioni semplici, onde seno e coseno, di varia frequenza ed ampiezza. Ottenendo lo spettro di potenza di una serie temporale di campioni EEG, le fluttuazioni di potenziale registrate da un elettrodo EEG istante per istante sono analizzate in sinusoidi di varia frequenza. Il quadrato del cosiddetto *coefficiente di Fourier* (l'ampiezza della sinusoide ad una particolare frequenza) a ciascuna frequenza è chiamato *potenza spettrale* di quella frequenza. Le variazioni

nella potenza spettrale di una frequenza in risposta a cambiamenti sperimentali del task o nel tempo possono rivelare relazioni tra l'attività di elaborazione di gruppi di neuroni e processi cognitivi, o tra gruppi di neuroni separati in diversi siti del cervello. L'elettroencefalogramma varia con l'attività, sia negli umani che in altri animali, e particolarmente con il ciclo sonno-veglia. [10]

Il contenuto spettrale del segnale EEG si estende fino a 100 Hz, sebbene la maggior parte del contenuto informativo sia al di sotto dei 40 Hz. Il contenuto armonico del segnale EEG viene suddiviso in 5 bande principali, denominate anche ritmi: ritmo alpha (α), ritmo beta (β), ritmo theta (θ), ritmo delta (δ), ritmo gamma (γ). Essi sono descritti nel seguito.

Ritmo alpha: onde con contenuto frequenziale tra 8 Hz e 13 Hz. Durante una registrazione EEG di superficie, le onde alpha hanno ampiezza approssimativamente tra 20 e 50 microvolt. Durante la registrazione effettuata direttamente dal tessuto cerebrale, mediante l'uso di un sottile elettrodo di metallo, il voltaggio del potenziale massimo è molto alto, perché vengono eliminate le interferenze e le dispersioni della scatola cranica e della pelle. Il ritmo alpha generalmente ha un andamento regolare e continuo anche se sono possibili variazioni individuali. Il ritmo alpha può essere registrato più precisamente da elettrodi posti nell'area posteriore del cranio, laddove è situata la regione encefalica, detta lobo occipitale, che riceve ed elabora i segnali provenienti dalla retina.

Considerando il rapporto univoco intercorrente tra l'apertura e la chiusura degli occhi e la comparsa del ritmo alpha, si è appurato che i fenomeni che concernono l'alpha vengono influenzati dagli stimoli che raggiungono il cervello passando attraverso la retina e il nervo ottico. Le onde alpha infatti sono predominanti nello stato di veglia rilassata ad occhi chiusi. Quando un individuo psichicamente e fisicamente in stato di riposo apre gli occhi il voltaggio del ritmo alpha si abbassa e alcune volte scompare del tutto. L'attenuazione delle onde alpha sembra essere correlata ad una maggiore disponibilità a ricevere uno stimolo sensoriale o un comando motorio. In molti individui il ritmo alpha può scomparire se questi intraprendono una attività mentale come ad esempio quella di fare calcoli [1,11]. Come la Tabella 1.1 illustra, lo stato mentale in cui si manifesta il ritmo alpha potrebbe anche essere uno stato attento: nelle regioni task-irrelevant per

l'inibizione di distrattori, e nelle regioni corticali task-relevant per una efficace allocazione delle risorse. Per esempio, le onde alpha nelle regioni cerebrali centrali e posteriori crescono con il crescere del carico mentale nei task di memorizzazione, riflettendo un crescente processo di inibizione di tali regioni, non richieste per lo specifico task, per l'aumentata necessità di sopprimere eventuali distrazioni [13].

Ritmo beta: onde con contenuto frequenziale tra 13 Hz e 30 Hz e bassa ampiezza, tra i 5 μ V e i 30 μ V. Viene normalmente registrato, al contrario del ritmo alpha, nelle regioni cerebrali frontali e centrali. Le onde del ritmo beta si identificano generalmente con uno stato di coscienza vigile e si manifestano durante attività mentale intensa, stati di allerta o stati di attenzione e concentrazione. Nel caso in cui, ad esempio, il paziente stia eseguendo dei calcoli si verifica frequentemente la sostituzione del ritmo alpha con il ritmo beta [1,11].

Ritmo theta: questo ritmo è caratterizzato da onde aventi un contenuto frequenziale tra 4 Hz e 7 Hz e ampiezza tra 2 μ V e 100 μ V. Il ritmo theta si presenta con irregolarità e viene quasi sempre registrato mediante elettrodi posti sopra i lobi temporali e parietali del cervello. Le onde theta sono predominanti durante le fasi di addormentamento e di sonno. Oggi, notevole interesse scientifico è rivolto alle onde theta in relazione a memoria e apprendimento [1,11].

Ritmo delta: la frequenza di questo ritmo elettroencefalografico varia da 0,5 Hz a 4 Hz e con ampiezza tra 20 μ V e 200 μ V. Si tratta di onde lente, che sono dominanti nei neonati; in individui normali la frequenza non è mai più bassa di due onde al secondo. Le onde delta si manifestano anche negli adulti nelle fasi di sonno profondo non-REM (cioè privo di Rapid Eye Movement) oppure in condizioni patologiche quali coma o forme tumorali. Questo ritmo si manifesta irregolarmente, in genere con onde isolate. Il ritmo delta di solito viene registrato mediante elettrodi collocati sopra i lobi temporali del cervello. L'attività delta varia da individuo a individuo e manifesta spesso grandi differenze anche nello stesso individuo se l'elettroencefalogramma o EEG è registrato in tempi diversi [1,11].

Ritmo gamma: onde con contenuto frequenziale superiore a 30 Hz, tipicamente di piccola ampiezza (inferiore a 20 μ V). Localizzate nelle regioni frontali del cervello, le onde gamma sono associate a fasi di elaborazione attiva e sono evidenti in attività cognitive di alto livello, che richiedono quindi

l'integrazione di informazioni elaborate in aree cerebrali diverse, quali riconoscimento di oggetti, recupero di informazioni dalla memoria, processi attentivi [1].

La Tabella 1.1 riassume le principali caratteristiche dei cinque ritmi EEG canonici. In particolare, l'ultima colonna elenca alcune tipologie di esperimenti (task) sottoponibili ad un soggetto al fine di valutare efficacemente la comparsa o la scomparsa del ritmo EEG corrispondente.

<i>Ritmo</i>	<i>Frequenza (Hz)</i>	<i>Ampiezza (μV)</i>	<i>Localizzazione</i>	<i>Stato mentale</i>	<i>Task</i>
Delta	0.5-4	20-200	Frontale e parietale	Sonno profondo non REM; condizioni patologiche; coma	Riconoscere uno specifico target in una serie di distrazioni
Theta	4-8	20-100	Temporale e parietale	Addormentamento; sonno non REM; stress emotivo; memoria ed apprendimento	Memorizzazione ed apprendimento
Alpha	8-14	20-50	Occipitale e parietale	Stato cosciente rilassato; occhi chiusi; occhi aperti, basso livello attentivo e di coinvolgimento;	Memorizzazione ed attenzione
Beta	14-25	5-30	Parietale e frontale	Coscienza vigile; concentrazione; attività mentale	Motori e cognitivi, esecuzione di calcoli
Gamma	>25	1-20	Frontale	Percezione cosciente; attenzione; integrazione informazioni	Mantenere diversi contenuti in memoria; riconoscere oggetti; richiamare contenuti dalla memoria

Tabella 1.1. Principali caratteristiche dei ritmi cerebrali, stato mentale a cui sono associati e possibili task per valutarne le variazioni.

La potenza spettrale alle varie frequenze cambia con l'età; la potenza alpha aumenta mentre i bambini crescono, mentre le potenze theta e delta decrescono. È grazie agli studi delle Neuroscienze cognitive che è stato possibile provare che i processi di memoria sono più legati ai cosiddetti *ritmi theta* e *gamma*, mentre invece l'attenzione sembra strettamente associata ai *ritmi alpha* e *gamma*. [10]

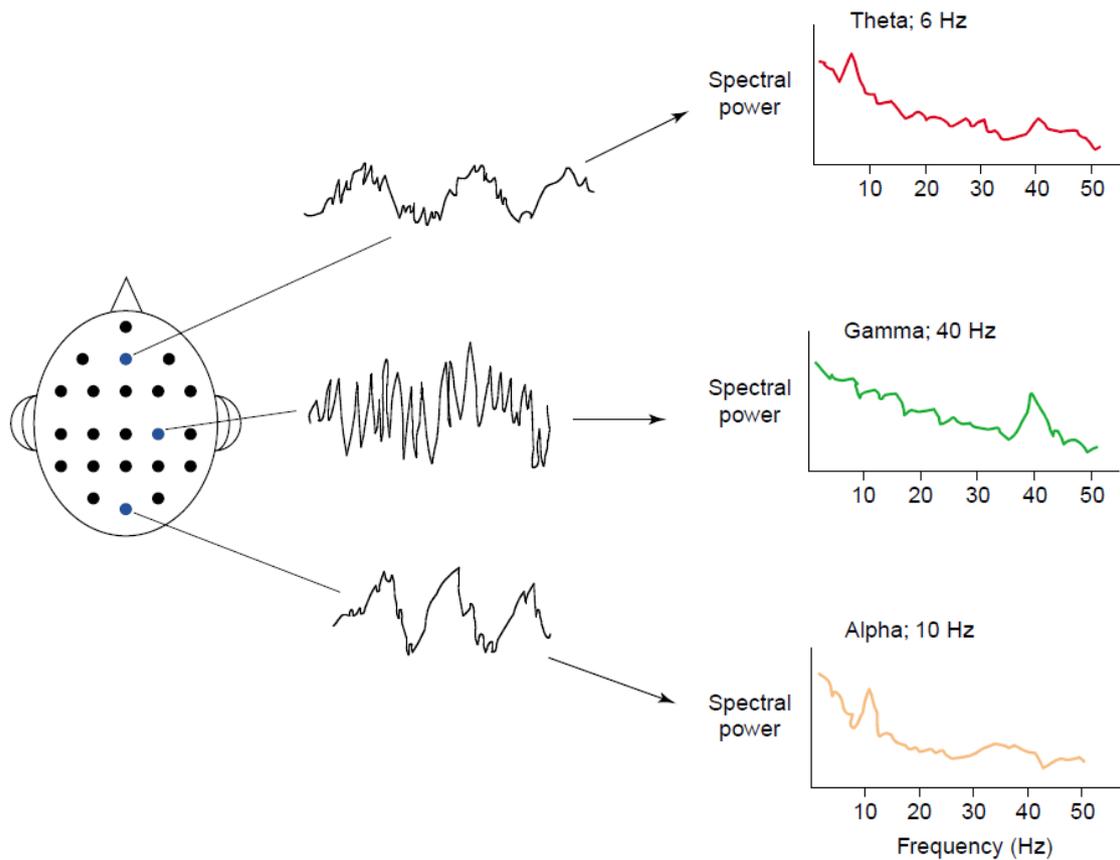


Figura 1.6. Esempi di alcuni spettri di potenza che mostrano picchi alle frequenze EEG canoniche (tratta da [10])

Classicamente, grazie al fatto che la potenza alpha era molto maggiore con gli occhi chiusi che con gli occhi aperti, si pensava che alpha riflettesse una situazione di cervello rilassato e sgombro. Un generale decremento della potenza alpha è stato associato ad una crescente richiesta di attenzione, allerta, e carico di attività in generale [12]. Come theta e gamma risultano essere importanti ritmi per i processi di memoria, così alpha e gamma paiono essere prominenti nell'attenzione. Sta diffondendosi la convinzione che le oscillazioni alpha indicano che l'attenzione sta attivamente sopprimendo l'attività corticale relativa a possibili distrattori come parte del processo di concentrare l'attenzione su importanti target. In altre parole, un incremento di ritmo alpha durante un task sarebbe associato ad un processo di inibizione di regioni corticali irrilevanti per lo specifico task richiesto, allo scopo di favorire l'allocazione di risorse attentive e il processing dell'informazione nelle regioni task-relevant [10]. Per esempio, la potenza alpha nelle regioni centro-

posteriori cresce con il crescere del carico mentale nei task di memorizzazione, riflettendo un crescente processo di inibizione di tali regioni, non richieste per lo specifico task, per l'aumentata necessità di sopprimere eventuali distrazioni [13].

Nella sua prima pubblicazione sull'EEG umano nel 1929, Berger arriva alla seguente conclusione: <<Anzi, credo di avere scoperto l'elettroencefalogramma di un uomo e di averlo qui pubblicato per la prima volta>>. Egli poi continua a descrivere l'EEG come una continuativa curva di costanti fluttuazioni, dalla quale si possono discernere due tipi di onde. Alcune onde più grandi, "del primo ordine", con una durata media di 90ms, e altre onde più piccole, "del secondo ordine", con una durata media di 35ms con deflessioni massime di 200uV. Nel suo secondo report (1930) chiamò quelle grandi *onde alpha* e quelle più piccole *onde beta*. Dopo la pubblicazione di ben 13 ulteriori articoli, fu solo nel 1934 che il suo lavoro ricevette la meritata riconoscenza della comunità scientifica [14].

Vi sono molti più processi cognitivi rispetto alle sole 5 bande di frequenza ben definite (delta, theta, alpha, beta, gamma), pertanto è ovvio che non si possa stabilire una corrispondenza 1:1 tra processi cognitivi da un lato ed oscillazioni EEG dall'altro. Ciò che è più probabile è che le oscillazioni EEG contribuiscono a differenti funzioni cognitive in dipendenza della regione cerebrale e dei parametri (ampiezza, frequenza, fase, coerenza) in oggetto. Regioni cerebrali diverse possono attuare diverse ma specifiche funzioni. Le oscillazioni a bassa frequenza rappresentano l'attività di cooperazione di reti neuronali di larga scala nel cervello, mentre le oscillazioni ad alta frequenza riflettono prioritariamente l'attività di popolazioni neuronali locali. Oscillazioni EEG coerenti in due regioni cerebrali distanti possono riflettere la cooperazione funzionale di tali due siti. Da ciò si evince come la cooperazione fra regioni corticali vicine fra loro possa essere riflessa da oscillazioni coerenti ad alta frequenza, mentre invece la cooperazione di parti distanti si riflette in oscillazioni coerenti a bassa frequenza [14].

1.5 ELETTROCARDIOGRAFIA

L'*elettrocardiogramma (ECG)* è la registrazione dei potenziali generati sulla superficie del corpo durante il processo di stimolazione della muscolatura cardiaca. Esso rappresenta uno dei test diagnostici più diffusi (oltre 200 milioni di ECG annui). Il segnale ECG gode infatti di alcune proprietà che ne fanno la principale indagine strumentale di primo livello: è spontaneo, acquisibile in modo semplice, non invasivo e a basso costo, e fornisce informazioni immediate sullo stato di salute di un soggetto. La registrazione dei potenziali elettrici cardiaci viene fatta tramite strumenti detti appunto *elettrocardiografi*, oggi diffusi in ogni reparto per i motivi sopra elencati. [1,15]

Così come i muscoli scheletrici, il muscolo cardiaco è stimolato elettricamente per contrarsi. Tale stimolazione è chiamata attivazione o eccitazione. Il muscolo cardiaco, detto *miocardio*, è carico elettricamente a riposo: l'interno della cellula è carico negativamente rispetto all'esterno. Se le cellule del miocardio, dette *cardiomiociti*, vengono stimulate, esse si depolarizzano, e il potenziale di riposo passa da negativo a positivo, e il muscolo si contrae. L'attività elettrica della singola cellula può essere registrata come potenziale d'azione. Mentre l'impulso elettrico si diffonde attraverso il cuore, il campo elettrico cambia continuamente in ampiezza e direzione. L'ECG è un grafico di questi segnali elettrici cardiaci. [4]

Il tracciato ECG può essere registrato con elettrodi posti direttamente sulla superficie cardiaca (*derivazione diretta*) o, più frequentemente, con elettrodi posti sulla superficie cutanea degli arti e del torace (*derivazione indiretta*).

La misura e l'analisi del potenziale elettrico sono basate sugli studi del medico fisiologo Willem Einthoven del 1906, secondo cui la conduzione elettrica cardiaca viene descritta con un vettore campo elettrico che varia in ampiezza e direzione periodicamente nel tempo. Le proiezioni di questo vettore lungo precise direzioni dello spazio, dette *derivazioni*, sono perciò dei potenziali elettrici. Una derivazione è una direzione che congiunge 2 elettrodi con un verso preciso. Nella sostanza acquisire il segnale ECG di diverse derivazioni significa vedere lo stesso fenomeno da diversi punti di vista. [15]

1.6 BASI CARDIOFISIOLOGICHE DEL SEGNALE ECG

È noto che una cellula cardiaca si depolarizza, inverte la sua polarità e si ripolarizza tramite movimento di ioni, e questo comporta un'oscillazione del potenziale all'interno della cellula da -85 mV di riposo a +25 mV. All'esterno della cellula invece, cioè sulla superficie del tessuto cardiaco o sulla cute, le oscillazioni di potenziale sono molto inferiori, dell'ordine del millivolt, a causa dell'attenuazione dovuta alla propagazione attraverso i tessuti. [15]

In superficie si registrano le differenze di potenziale nel campo elettrico creato dalla depolarizzazione cellulare durante la propagazione dell'eccitamento, come lo spostamento di un'onda il cui fronte divide la zona attiva depolarizzata (esternamente carica negativamente) da quella inattiva polarizzata (esternamente carica positivamente). La figura seguente è esplicativa di tale concetto. [16]

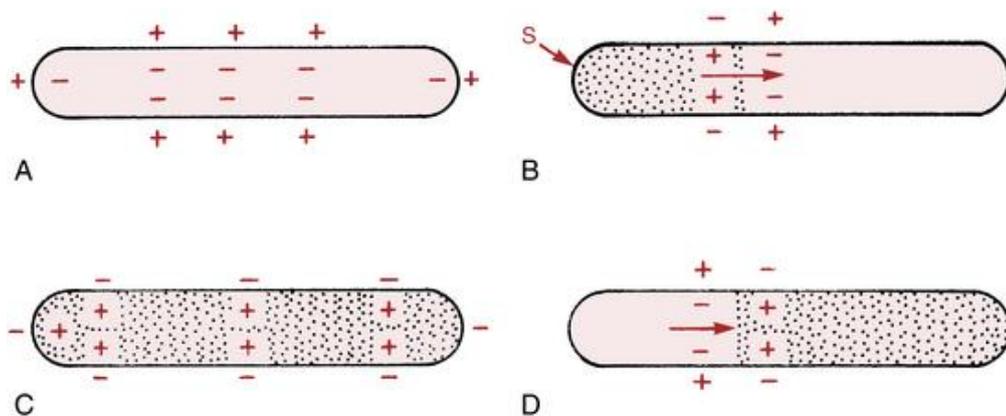


Figura.1.7. Depolarizzazione e ripolarizzazione. A: la cellula cardiaca a riposo è polarizzata, ovvero è caricata elettricamente, negativamente all'interno e positivamente all'esterno. B: quando la cellula viene stimolata (S), essa inizia a depolarizzarsi (area a puntini). C: la cellula totalmente depolarizzata è carica positivamente all'interno e negativamente all'esterno. D: la cellula torna allo stato di riposo, ripolarizzandosi. (tratta da [16])

La depolarizzazione (stimolazione) degli atri produce l'onda P sull'ECG. Segue una breve fase isoelettrica in cui il potenziale è nullo e corrisponde all'arrivo dell'impulso sul piano Atrio-ventricolare. La depolarizzazione dei ventricoli produce il complesso QRS, con i punti Q ed S a potenziali negativi (iperpolarizzazione), e in R il picco massimo (1-2mV). Segue una breve fase isoelettrica dovuta al ritardo di propagazione dalle cellule di conduzione a quelle muscolari contrattili. La ripolarizzazione dei ventricoli produce il complesso ST-T. [16]

Le diverse derivazioni di un ECG forniscono "visuali" elettriche diverse e danno informazioni circa le differenti regioni del cuore. Infatti, le polarità e le ampiezze delle onde variano a seconda della derivazione esaminata. Per questo motivo quando si deve interpretare un tracciato elettrocardiografico è fondamentale sapere a quale derivazione si fa riferimento. La Fig. 1.8, ad esempio, mostra il tracciato ECG acquisito in derivazione bipolare periferica I, detta D1, che corrisponde alla configurazione di due elettrodi sui polsi e un elettrodo alla caviglia destra (che serve esclusivamente come riferimento). [4,15]

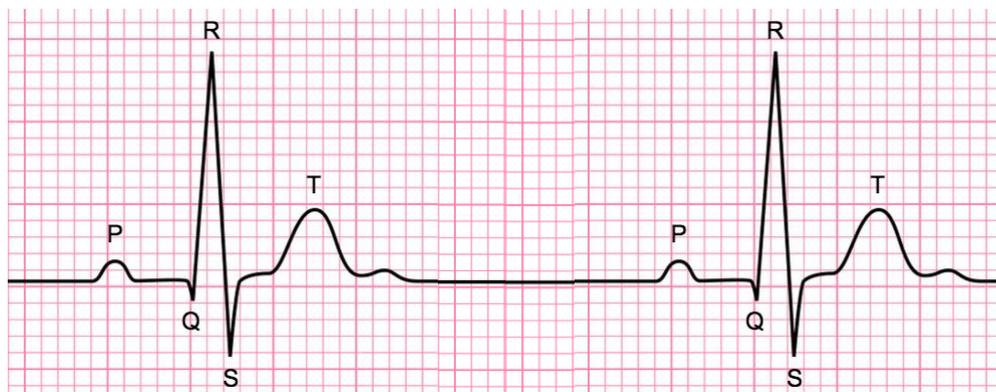


Figura 1.8. Esempio di tracciato ECG in derivazione D1. Sono evidenti le onde e i complessi canonici P, QRS, T e le fasi isoelettriche (tratta da [16])

1.7 HEART RATE (HR) E HEART RATE VARIABILITY (HRV)

I parametri caratteristici estraibili da un ECG sono numerosi, primo fra tutti la distanza temporale tra i due picchi R-R di battiti successivi, sul cui valore si basa il calcolo della frequenza cardiaca (*HR, Heart Rate*). La HR è in genere abbastanza costante, ma esiste un valore della sua variabilità, tipicamente di 5%, al di sopra del quale si tende a diagnosticare *aritmia cardiaca*. Altri parametri importanti sono gli intervalli PQ, QRS, QT ed i segmenti PQ e ST. [15]

Quando il tracciato ECG non presenta aritmie e la HR è compatibile con quella fisiologica (60-100bpm), il ritmo cardiaco è detto *sinusale*, poiché in condizioni fisiologiche è il *nodo senoatriale* a comandare la conduzione cardiaca. [15]

In un organismo sano, vi è un equilibrio dinamico tra sistema nervoso simpatico (SNS) e sistema nervoso parasimpatico (SNP), le due branche del sistema nervoso autonomo. L'attività del SNP prevale a riposo, risultando in una HR media di 75 bpm. Questa è significativamente più bassa della frequenza intrinseca del nodo SA, che decresce con l'età da una media di 107 bpm a 20 anni ad una media di 90 bpm a 50 anni [17]. La branca parasimpatica è capace di rallentare il cuore a 30 o 20 bpm, e addirittura di fermarlo quasi del tutto [18].

La frequenza cardiaca calcolata ad un dato tempo rappresenta l'effetto netto della risposta neurale dei nervi parasimpatici, che rallentano HR, e dei nervi simpatici, che la accelerano. In letteratura, ci si riferisce ad un simile meccanismo come meccanismo *antagonista*. [19]

Grazie alle odierne tecniche di elaborazione dei segnali si possono acquisire continue serie temporali di dati da processi fisiologici. Ciò permette di vedere come i processi biologici varino in modo complesso e non lineare, anche durante condizioni di stato stazionario.

Ad esempio, oggi sappiamo che il normale ritmo sinusale del cuore a riposo è altamente irregolare durante condizioni di stato stazionario: *un cuore sano non è un metronomo*. Il cambiamento negli intervalli temporali fra battiti adiacenti è denominato *Heart Rate Variability (HRV)*. [19]

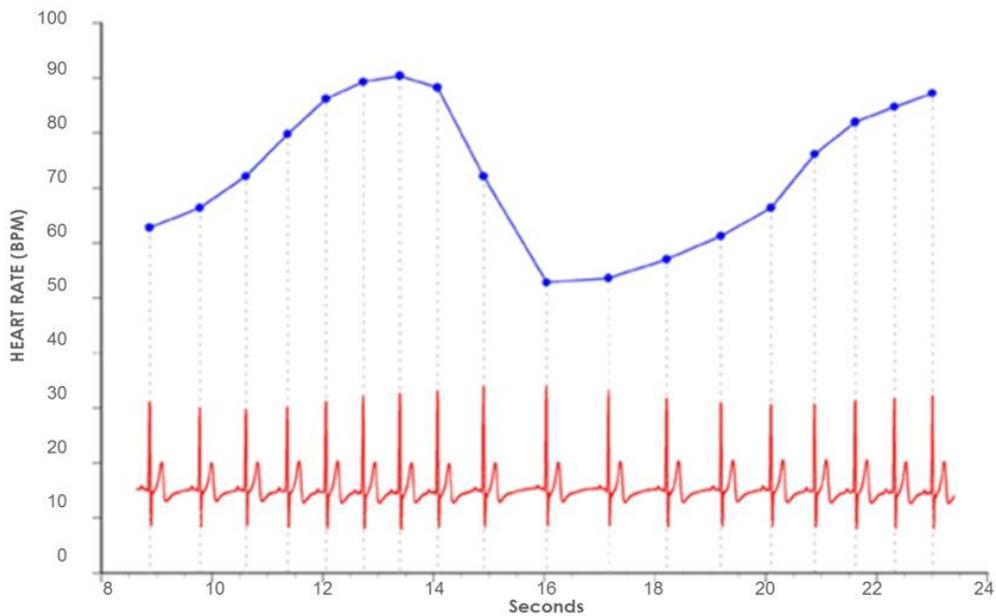


Figura 1.9. Heart Rate Variability ed ECG. Notare come l'intervallo temporale fra picchi R-R successivi non rimane costante, bensì varia. (tratta da [19])

La variabilità del battito cardiaco è molto evidente quando la HR è esaminata da battito a battito, ma diviene trascurabile quando si calcola un valor medio nel tempo. Tali fluttuazioni nella frequenza cardiaca sono il risultato di interazioni complesse e non lineari fra sistemi fisiologici diversi. Le interazioni fra attività nervosa autonoma, pressione sanguigna, e sistemi di controllo respiratori producono ritmi di breve durata nelle misure di HRV. La forma più comune per osservare questi cambiamenti è il cosiddetto *tacogramma*, il grafico di una sequenza di intervalli temporali fra le onde R (vedi Figura 1.10). [19]

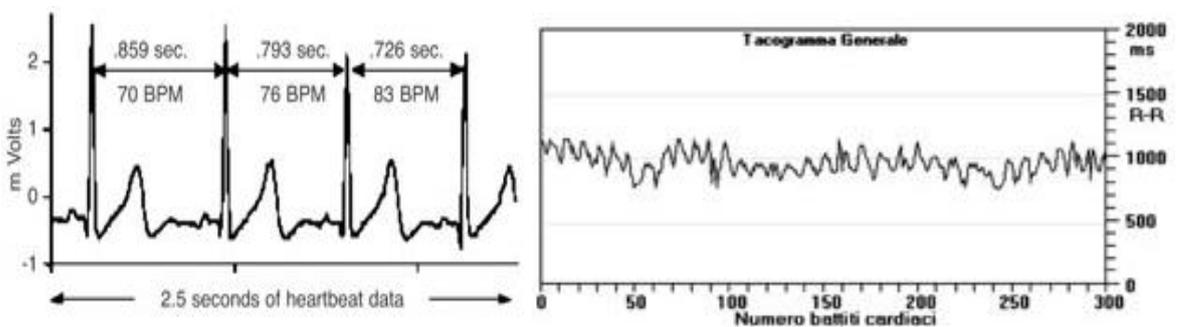


Figura 1.10. A sinistra è mostrato l'elettrocardiogramma relativo ad una acquisizione di 2,5s. A destra, il Tacogramma corrispondente; (tratto da [20])

1.8 IMPORTANZA DELLA HRV E SUOI METODI DI ANALISI

Un livello ottimale di variabilità all'interno dei sistemi di regolazione primari di un organismo è fondamentale per la flessibilità e adattabilità o resilienza, parametri di benessere e corretto funzionamento. Una bassa HRV è stata confermata essere premonitrice di futuri problemi cardiaci. Una ridotta HRV è stata osservata anche in pazienti affetti da ansia e depressione. Dagli studi effettuati da Berntson e colleghi nel 2008 [21], si evince che una ridotta HRV può essere correlata con malattie e mortalità perché essa riflette una ridotta capacità di regolazione, che è la capacità di rispondere adeguatamente alle situazioni di esercizio o stress. In quest'ottica, infatti, i pazienti con HRV bassa dimostrano una ridotta capacità regolatoria cardiaca e una elevata probabilità di infarto miocardico primario. [19]

La Heart Rate Variability è indicatore anche della resilienza psicologica e della flessibilità comportamentale, riflettendo la capacità di un individuo di adattarsi efficacemente a contesti sociali ed ambientali che cambiano [21]. Recentemente, diversi studi hanno mostrato l'associazione fra alti livelli di HRV a riposo e performance in task cognitivi che richiedevano l'uso di funzionalità esecutive, e che la HRV può essere innalzata per produrre miglioramenti nelle funzioni cognitive. [19]

La Heart Rate Variability può essere approcciata con diversi tipi di analisi: l'analisi nel dominio delle frequenze o analisi spettrale di potenza (PSD), e l'analisi nel dominio dei tempi. In entrambi i metodi, si determinano in primis gli intervalli di tempo fra complessi QRS normali successivi. Tutti i battiti irregolari (battiti atipici) non generati dalla depolarizzazione indotta dal nodo senoatriale vengono eliminati nella estrazione del tacogramma ed eventualmente sostituiti mediante appropriata interpolazione tra battiti precedenti e successivi. [19]

Possiamo usare l'analisi spettrale di potenza per separare la HRV nei suoi ritmi componenti che operano a diversi intervalli di frequenze. L'analisi PSD fornisce informazioni su come la potenza del segnale HRV è distribuita (in termini di varianza e ampiezza di un dato ritmo) in funzione della frequenza. Il più grande vantaggio dell'analisi spettrale rispetto alle misurazioni nel dominio dei tempi è che essa garantisce sia informazioni di frequenza sia di ampiezza degli specifici ritmi che esistono nella forma d'onda HRV, fornendo un metodo per quantificare le varie

oscillazioni su ogni dato periodo nella registrazione dell'HRV. I valori sono espressi come PSD (unità di misura del segnale al quadrato diviso Hz), e pertanto la potenza all'interno di una certa banda di frequenze si ottiene calcolando l'area sottesa dalla curva su quell'intervallo di frequenza. La potenza o altezza di ciascun picco a ciascuna specifica frequenza indica l'ampiezza del ritmo a quella frequenza. [19]

1.9 BANDE DI POTENZA LF ED HF DELL' HRV

Esistono quattro principali bande di frequenza dei ritmi cardiaci: high frequency (HF), low frequency (LF), very-low frequency (VLF) e ultra-low frequency (ULF). Nel seguito si daranno alcuni cenni sulle prime due classi, le più importanti per caratterizzare fenomeni e processi che si sviluppano nell'arco di minuti o alcune decine di minuti (come sono le tempistiche degli esperimenti qui esaminati). Fenomeni nelle bande VLF e ULF richiedono registrazioni di più lunga durata per emergere. [19]

La potenza HF è la potenza del segnale nella banda 0.15-0.4 Hz. Essa riflette l'attività parasimpatica o vagale (da *nervo vago*) ed è spesso chiamata *banda della respirazione* perché corrisponde alle variazioni di frequenza cardiaca relative al ciclo respiratorio: l'HR accelera durante l'inspirazione e rallenta durante l'espiazione. Infatti, durante l'inspirazione il centro cardiovascolare inibisce l'attività vagale e ciò fa aumentare la HR. Dualmente, durante l'espiazione, il centro cardiovascolare ripristina l'attività vagale e ciò rallenta il cuore grazie al rilascio di acetilcolina [22]. Una bassa attività ad alta frequenza è riscontrabile in pazienti cardiopatici e in pazienti sotto stress o che soffrono di panico, ansia o paura. [19]

La potenza LF è la potenza del segnale nella banda 0.04-0.15 Hz. Essa riflette l'attività a riposo dei barocettori, cioè i recettori neurosensoriali che rispondono a variazioni pressorie del sistema cardiovascolare. Per questo motivo tale range di frequenze è detto anche *banda dei barocettori* [23]. Quando la pressione sanguigna cresce, i tessuti carotidei ed aortici si distendono, perciò aumenta l'attivazione dei barocettori, che iniziano a generare potenziali d'azione con maggiore frequenza. I

barocettori influenzano sia l'attività simpatica (l'attività simpatica viene ridotta in seguito ad una aumentata scarica dei barocettori e viceversa) sia l'attività parasimpatica (l'attività parasimpatica aumenta in seguito ad una aumentata scarica dei barocettori e viceversa). L'effetto dei barocettori sul sistema nervoso autonomo (sia simpatico che parasimpatico) ha l'obiettivo di mantenere la pressione arteriosa entro stessi limiti. In conclusione, la potenza in banda LF riflette meccanismi vagali, simpatici e barocettivi. [19]

Il rapporto LF/HF è il rapporto fra potenza LF e potenza HF. Secondo la teoria dell'equilibrio autonomo i sistemi nervosi *simpatico* e *parasimpatico* agiscono in modo antagonista regolando l'attivazione del nodo senoatriale, dove l'aumento dell'attività di uno corrisponde ad una diminuzione dell'attività nell'altro. Così, il rapporto LF/HF è interpretabile come indice simpato-vagale. Infatti, dal momento che la banda HF riflette l'attività del solo *simpatico*, mentre la banda LF tiene conto sia del *simpatico* che del *parasimpatico*, una diminuzione di questo rapporto è in genere vista come una sbilanciamento verso l'attività parasimpatica, mentre un suo incremento è indice di uno sbilanciamento verso l'attività simpatica, come si può vedere nei soggetti a cui si presenta un compito che richiede sforzo ed aumentata attivazione del *sistema nervoso simpatico*. [19]

Poiché risposte emotive, attentive, di stress coinvolgono anche il sistema nervoso autonomo (oltre al sistema nervoso centrale) esse possono essere associate a variazioni nell'HRV e in particolare a variazioni nel bilancio simpato-vagale. Ad esempio, in condizioni di un aumento del carico di lavoro (condizioni di stress) si ha un incremento della frequenza cardiaca, e un aumento del rapporto fra potenza nella banda a basse frequenze e potenza nella banda ad alte frequenze dell'HRV. Inoltre, lavori recenti hanno mostrato un incremento del rapporto LF/HF anche durante immersione in realtà virtuale. [24]

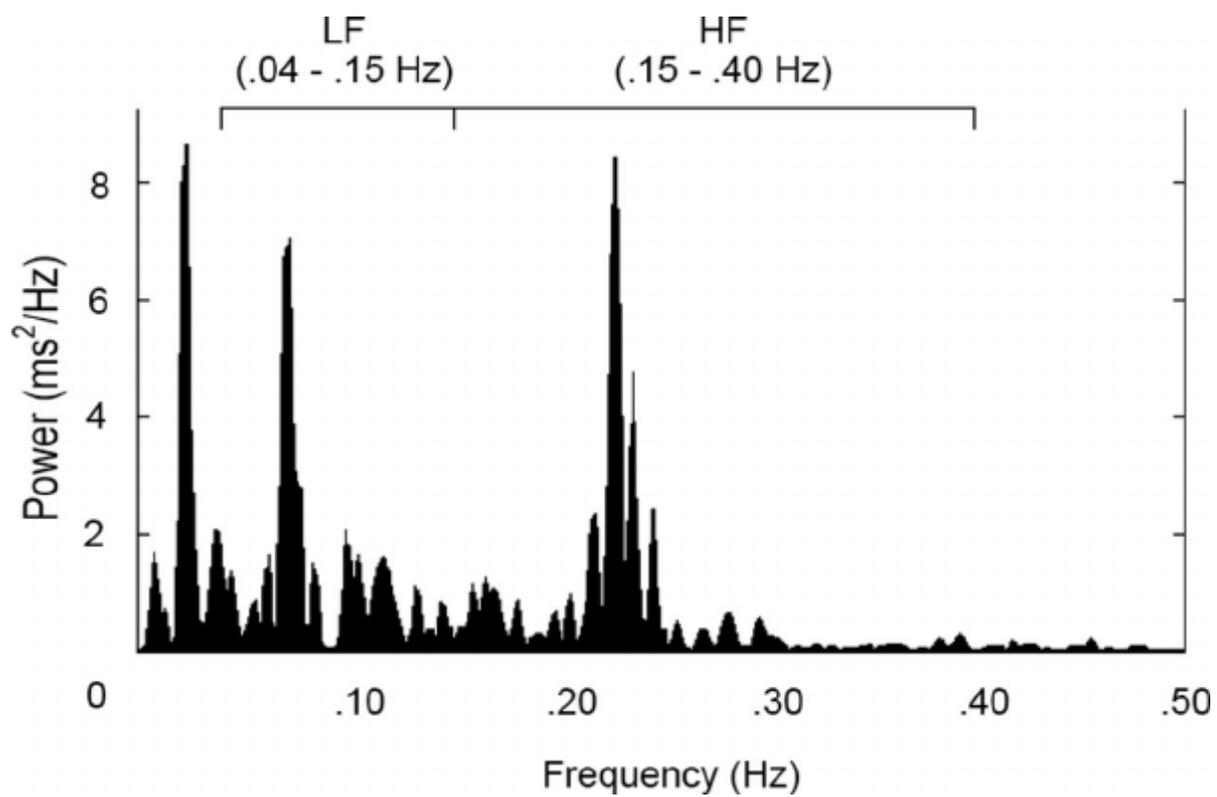


Figura.1.11. Spettro di potenza del segnale HRV. In evidenza le bande LF ed HF; (tratta da [25]).

CAPITOLO 2

DESCRIZIONE DELLE ESPERIENZE IN LABORATORIO PER L'ACQUISIZIONE DEI SEGNALI EEG ED ECG

All'interno del vasto progetto descritto nel primo capitolo, l'obiettivo specifico del presente lavoro di tesi è stato quello di realizzare una interfaccia grafica utente *GUI* (*Graphical User Interface*) in MATLAB®. Tale GUI è stata sviluppata per essere testata ed applicata su segnali acquisiti durante due diverse esperienze, svolte presso il Laboratorio di Realtà Virtuale nel Campus di Forlì (Università di Bologna) e presso il Laboratorio di Ingegneria Biomedica nel Campus di Cesena (Università di Bologna). In questo capitolo si darà una descrizione delle suddette esperienze e dei diversi sistemi di acquisizione utilizzati, *OpenBCI* e *Neurowave*, nonché del setup strumentale e dei passi seguiti. Per quanto riguarda il ruolo specifico da me ricoperto in queste esperienze di laboratorio, si evidenzia che – insieme ad altri due tesisti (uno triennale e uno magistrale) - ho partecipato attivamente alle acquisizioni su alcuni dei soggetti coinvolti nell'esperimento. La tesista magistrale ha partecipato a tutte la acquisizioni fornendomi i rimanenti segnali EEG ad ECG.

2.1 IL LABORATORIO DI REALTÀ VIRTUALE E L'AMBIENTE "CAVE"

Per le esperienze a Forlì abbiamo usufruito del *Laboratorio di Realtà Virtuale* che, fondato nel 2001, è tutt'ora parte del CIRI (Centro Interdipartimentale per la Ricerca Industriale) AERONAUTICA SPAZIO E MOBILITA' dell'Università di Bologna – Campus di Forlì.

Le attività svolte quotidianamente all'interno del Laboratorio si concentrano sull'integrazione di tecnologie quali CAD (Computer Aided Design), CAID (Computer Aided Industrial Design), Realtà Virtuale, Realtà Aumentata e Ingegneria Inversa. Inoltre, negli ultimi anni il laboratorio ha acquisito esperienza nella progettazione, lo sviluppo e la sperimentazione di interfacce uomo-macchina innovative. [26] Fra le diverse attrezzature e tecnologie di cui dispone il laboratorio, è stato utilizzato il cosiddetto *CAVE*. Il *CAVE* (*Cave Automatic Virtual*

Environment) è un ambiente virtuale immersivo riconfigurabile. Si ispira ai sistemi virtuali interattivi che furono inizialmente concepiti per l'esplorazione di modelli tridimensionali di grandi dimensioni in scala 1:1. Tre proiettori di ultima generazione (vedi Figura 2.2) illuminano tre schermi di dimensioni 2,5 m x 1,9 m, ottenendo una superficie totale di proiezione pari a circa 14 m² (vedi Figura 2.1). Il sistema supporta la visualizzazione stereoscopica attiva. [26]



Figura. 2.1. Laboratorio CAVE, presso la sede di via Fontanelle 40 Forlì, al piano terra. (tratta da [26])



Figura. 2.2. Tre proiettori di ultima generazione retroproiettano sui tre schermi di dimensioni 2,5 m x 1,9 m. (tratta da archivio personale)

2.2 FINALITÀ DELL'ESPERIMENTO NEL "CAVE"

Le acquisizioni sperimentali nel CAVE di Forlì si inseriscono in un progetto finanziato dalla Comunità Europea dal titolo "CAbin Systems design Toward passenger wellbEing" (CASTLE).

L'obiettivo ultimo del progetto è la realizzazione di abitacoli di aerei che massimizzino il 'benessere' (in termini di comodità e funzionalità) dei passeggeri, ovvero una progettazione 'human-centered'.

A tale scopo, tra le altre cose, il progetto si pone l'obiettivo di investigare se, attraverso misure di parametri fisiologici, sia possibile ottenere valutazioni oggettive del livello di comfort, gradimento, coinvolgimento di un soggetto all'interno della cabina passeggeri di un aereo e se cabine differenti (ad esempio nei materiali, colori, organizzazione degli spazi, etc.) possano dare origine a differenze nei parametri fisiologici misurati (sottintendendo dunque effetti differenti sul soggetto).

Nello specifico, i ricercatori di Forlì hanno realizzato due ambienti virtuali di cabine passeggeri (che differiscono per alcuni aspetti, cabina B1 e cabina B2), e le prove sono state concepite per investigare lo stato di eccitazione/gradimento/attenzione elicitato da ciascuna cabina in soggetti 'virtualmente' immersi nella cabina stessa.

L'architettura interna delle due ambientazioni era strutturalmente identica: un divano, un tavolo con TV a schermo piatto, comode poltrone, corridoio per accedere al bagno e alle altre stanze. A cambiare erano fondamentalmente colori e materiali delle forniture, del pavimento e degli arredi.

Mentre la cabina B2 era caratterizzata da tavoli e pianali in marmo nero, interni e dettagli di un colore azzurrino tenue, poltrone beige e pavimento monocromatico grigio, la cabina B1 presentava legno marrone al posto del marmo, pavimento con una trama geometrica e colori più accesi rispetto alla prima cabina. In Figura 2.3 sono mostrate le due ambientazioni virtuali sopra descritte.



Figura 2.3. Ambientazioni virtuali di cabine passeggeri: cabina B2 (in alto) e cabina B1 (in basso); (tratte da archivio personale)

Lo scopo di questa esperienza è analizzare i tracciati EEG ed ECG dei soggetti partecipanti, e valutare se l'interazione con tali due ambienti virtuali (B1 e B2) dà origine a variazioni in questi segnali (principalmente per quanto riguarda la loro distribuzione di potenza tra le varie bande) che possano fornire informazioni oggettive circa lo stato di gradimento/eccitazione/coinvolgimento scaturito in ciascun soggetto per effetto dei colori, materiali e geometrie impiegati in ciascuna cabina. Infatti, come descritto nel capitolo precedente, variazioni nella potenza del segnale EEG specialmente in banda *alpha*, e variazioni nella potenza del segnale di HR tra le bande LF e HF, sono legate a stati di eccitazione/coinvolgimento. Quello che è stato ricercato dunque in questa esperienza è valutare se si riuscisse a

riscontrare una differenza significativa delle potenze in banda *alpha* del segnale EEG e delle potenze in banda LF e HF del segnale di HR nelle varie fasi dell'esperimento e nelle due cabine. Di interesse anche il fatto che ai soggetti veniva richiesto di compilare un questionario alla fine dell'esperienza, da cui è possibile derivare una valutazione soggettiva di gradimento delle due cabine che può dunque essere confrontata con le valutazioni oggettive scaturite dall'analisi dei segnali EEG e HR. Da un'analisi preliminare sui dati acquisiti, sembra che quando il soggetto è più coinvolto dalla cabina (quindi è più attento e più sollecitato visivamente), la potenza *alpha* nel segnale EEG sia più bassa.

Il grado di immersione nella realtà virtuale è stato soddisfacente ed adeguato allo scopo, grazie a più fattori: la leggera inclinazione dei due megaschermi laterali verso il lato lungo del palco, così da rendere più veritiera l'idea di "corridoio" (vedi Figura 2.4); la presenza in sala del classico rumore di fondo udibile all'interno di un aereo in volo; la presenza dell'avatar virtuale che riproduceva i movimenti di mani, busto e piedi del soggetto reale grazie al meccanismo di tracking implementato dal sistema *kinect* annesso; ed infine, il rumore riscontrabile all'impatto dell'avatar con gli oggetti durante l'interazione.

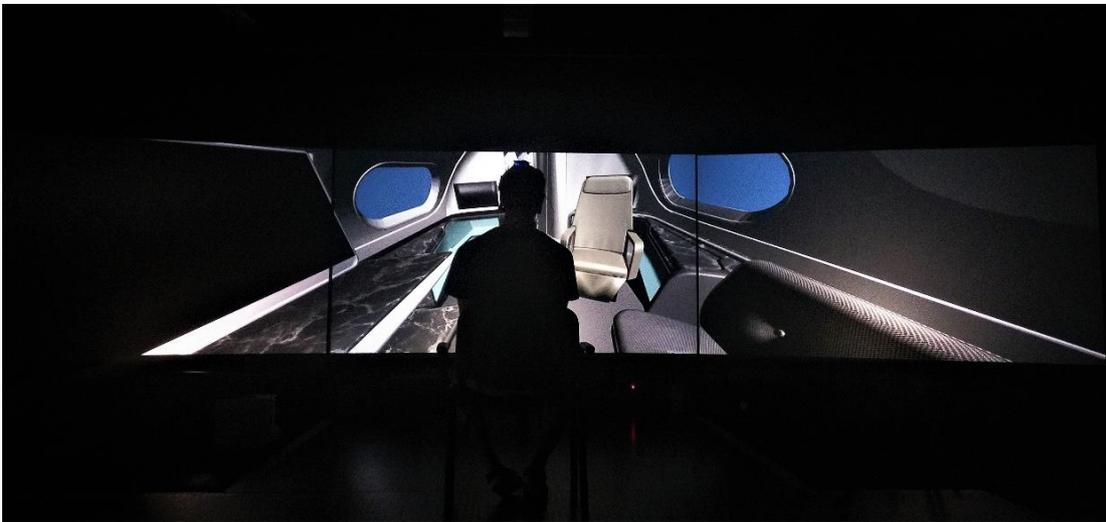


Figura. 2.4 Uno dei soggetti volontari durante l'esperimento. Nello specifico: cabina B2. (tratta da archivio personale)

2.3 FASI DELL'ESPERIMENTO NEL CAVE

Complessivamente 30 volontari sani hanno partecipato all'esperimento. I soggetti partecipanti sono stati reclutati tra gli studenti dei Corsi di Laurea in Ingegneria Aerospaziale a Forlì (Università di Bologna) e tra gli studenti dei Corsi di Laurea in Ingegneria Biomedica a Cesena (Università di Bologna). Come precedentemente indicato io ho partecipato attivamente alle acquisizioni sui primi 7 soggetti; le successive acquisizioni sono state portate avanti dalla tesista magistrale

Un soggetto volontario che si presta ad un esperimento in realtà virtuale su un palco esteso e con tecnologie che possono essere anche avanzate, necessita di una fase di addestramento (*training*), grazie alla quale egli può famigliarizzare con la strumentazione della prova. Nella fattispecie, ciascun soggetto veniva fatto salire sul palco senza indossare la strumentazione di acquisizione (la cuffia con gli elettrodi EEG e i due elettrodi ECG). Qui gli si presentava una delle due ambientazioni virtuali (quella che sarebbe poi stata presentata per prima nelle successive fasi di acquisizione), e gli si spiegava come muoversi nella stanza e come essere correttamente tracciato dal sistema *kinect*. Terminata la fase di *training*, si procedeva con la vera e propria acquisizione.

Per prima cosa ai volontari veniva chiesto di togliere orecchini e collanine, se presenti. A questo punto iniziava la fase di "vestizione" del soggetto, ossia la procedura di preparazione, cutanea e strumentale, al fine di permettere la migliore acquisizione possibile una volta sul palco del CAVE. Questa fase generalmente veniva fatta in coppie, per ottimizzare le tempistiche.

Mentre un operatore posizionava la cuffia per EEG sulla testa del soggetto, l'altro si preparava ad applicare i due elettrodi per ECG sulla cute di questo, leggermente sotto le clavicole, per mezzo di un cerotto adesivo. Sia gli elettrodi per ECG sul petto sia quelli per EEG sullo scalpo dovevano sempre essere riempiti con un gel conduttore particolare, in modo da minimizzare l'impedenza d'elettrodo e garantire una maggiore adesività di questo sulla cute. Ai soggetti veniva poi trattata la pelle dei lobi delle orecchie tramite pasta sgrassante, sui quali applicare poi due elettrodi. Il tema dei gel conduttivi e paste sgrassanti viene trattato alla fine del capitolo.

Terminata la fase di preparazione del soggetto, lo si conduceva sul palco rialzato del laboratorio CAVE. Come risulta chiaro dalla Figura 2.4, una sedia era posizionata al centro del palco, in posizione ottimale per godere, da seduti, della migliore visuale sui tre megaschermi.

Per ciascuna delle due ambientazioni virtuali (cabine B1 e B2), l'esperimento si componeva di quattro fasi ciascuna di 5min, per un totale di 20min per cabina, a cui andava sommato il tempo necessario alla pulizia di cuffia ed elettrodi, per il successivo volontario.

1) Fase di rilassamento con VR spenta

Il soggetto, seduto comodamente sulla sedia in dotazione, vedeva di fronte a sé tre schermi neri (Realtà Virtuale off). L'intero laboratorio era buio, a luci spente. Si chiedeva al soggetto di rimanere in quella posizione per 5min, di rilassarsi, liberando la mente, e tentando di limitare al massimo i movimenti del corpo, ad eccezione, ovviamente, dei movimenti della cassa toracica per la respirazione.

2) Fase di rilassamento con VR accesa

Questa fase ricalca esattamente la precedente, con l'unica differenza che questa volta il sistema era acceso e davanti agli occhi del soggetto si presentava l'ambientazione virtuale dell'abitacolo del velivolo. In questa fase il soggetto, dunque, era libero di esplorare con lo sguardo la cabina, restando però seduto e rilassato.

3) Fase di interazione con l'ambiente virtuale

Al soggetto veniva chiesto di alzarsi dalla sedia (che veniva prontamente prelevata e fatta uscire dall'area del palco); in questo modo il sistema *kinect* rilevava la presenza di un soggetto nel suo campo di azione, lo riconosceva, ed iniziava il suo tracking motorio e facciale. Ogni suo movimento reale era riprodotto dall'avatar virtuale nell'ambientazione. Iniziavano quindi i minuti di vera interazione, in cui il volontario camminava nel palco (un passo reale corrispondeva a tre passi virtuali), voltandosi ed abbassandosi per raggiungere gli oggetti, ed esplorare così l'intero

volume della cabina. In questa fase è stato critico il tema degli artefatti da movimento.

4) Fase di rilassamento con VR accesa post-interazione

Una volta riposizionata la sedia sul palco, al soggetto veniva chiesto di accomodarsi. La VR rimaneva accesa di fronte ai suoi occhi, ma nessuna ulteriore interazione era consentita (veniva disabilitato il sistema di face-tracking per evitare possibili artefatti). In tale posizione, il soggetto si rilassava e liberava la mente.

Ogni soggetto era dunque sottoposto a ciascuna di queste quattro fasi, prima per una cabina poi per l'altra cabina. L'ordine di presentazione delle cabine è stato controbilanciato tra i partecipanti. Alla fine delle due acquisizioni (complessivamente 40 minuti) i soggetti erano sottoposti a un questionario per una valutazione soggettiva delle due cabine.

L'articolazione nelle varie fasi aveva i seguenti scopi:

- a) Verificare se l'immersione nella realtà virtuale anche in condizioni di rilassamento determina modificazioni nei segnali EEG e di HR (confronto dei segnali tra la fase 1 e le fasi 2 e 4).
- b) Verificare se l'interazione con la realtà virtuale (nella fase 3) modifica i segnali rispetto alla fase precedente di immersione senza interazione (confronto dei segnali tra fase 2 e fase 4).
- c) Verificare differenze nei segnali in ciascuna fase nel caso di acquisizioni relative alla cabina B1 e nel caso di acquisizioni relative alla cabina B2 (confronto tra fasi corrispondenti nelle due acquisizioni).

Si sottolinea che non sono state fatte analisi dei segnali durante la fase 3 a causa dell'elevato livello di corruzione dei segnali causato dai movimenti del soggetto durante tale fase.

Qui di seguito vengono riportati sinteticamente i risultati preliminari ottenuti dall'analisi dei segnali. Si precisa che questi risultati sono riportati per completezza ma tali analisi e la valutazione dei risultati non sono state condotte nello specifico in questa tesi (bensì si tratta di un lavoro condotto specificamente – oltre ad altro – da una tesista magistrale). Infatti, questo lavoro di tesi è stato finalizzato allo

sviluppo di un'interfaccia grafica che consenta – attraverso una interfaccia user-friendly – la visualizzazione e l'analisi spettrale di segnali EEG ed ECG. I segnali acquisiti nelle esperienze qui illustrate sono stati utilizzati per lo sviluppo e il test di tale interfaccia, mentre una valutazione rigorosa e dettagliata dei risultati delle analisi di tali segnali esula dallo scopo di questo lavoro.

I risultati delle analisi sui segnali acquisiti in questa esperienza hanno messo in evidenza che:

- La potenza in banda alpha dei segnali EEG (specialmente nelle regioni occipitali) diminuisce durante l'immersione nella realtà virtuale (fasi 2 e 4), rispetto alla condizione di realtà virtuale spenta (fase 1), in entrambe le cabine. Questo indica che la presentazione dell'ambiente virtuale determina uno stato di attenzione/eccitazione nel soggetto anche solo attraverso una sollecitazione visiva ed esplorazione visiva.

- La potenza in banda alpha dei segnali EEG resta sostanzialmente inalterata tra la fase 2 e la fase 4, indicando che l'interazione con la realtà virtuale che avviene nell'esperienza non introduce modificazioni significative, sia nel caso della cabina B1 che nel caso della cabina B2.

- La potenza in banda alpha diminuisce maggiormente passando dalla fase 1 alla fase 2 nel caso di presentazione della cabina B2 rispetto al caso di presentazione della cabina B1. Questo sembra suggerire che la cabina B2 elicit un maggiore stato di attenzione/eccitazione rispetto alla cabina B1, che può anche essere imputato alla maggiore luminosità e vivacità di colori della cabina B2 rispetto alla B1. Interessante è il fatto che dal questionario compilato risulta che la maggior parte dei soggetti abbia gradito maggiormente la cabina B2 rispetto alla B1 (la cabina B2 li ha maggiormente coinvolti/sollecitati), in coerenza con il risultato dell'analisi oggettiva sui segnali.

- Infine, dall'analisi dei segnali è stato anche rivelato che il gradimento di una o dell'altra ambientazione virtuale si riscontrerebbe già nella fase di training iniziale. Infatti, sembra che un soggetto che svolga l'addestramento con la cabina B2, che è quella più luminosa, inizi l'acquisizione con la potenza *alpha* più bassa (già nella fase di rilassamento con VR spenta), a prescindere dalla preferenza; mentre coloro

che iniziano con la cabina B1, che ha toni più scuri e tendenti al blu, ha un *alpha* più elevato. Tale questione potrebbe essere legata all’impatto visivo dei colori e della luminosità dell’ambiente, che rimangono “impressi” nella mente del soggetto.

2.4 SETUP STRUMENTALE UTILIZZATO NELL’ESPERIMENTO NEL ‘CAVE’: IL SISTEMA DI ACQUISIZIONE “OPEN BCI”

La *Cyton OpenBCI Board* è una interfaccia neurale ad 8 canali, Arduino-compatibile, con processore a 32-bit. Al suo interno, la Cyton Board implementa il microcontrollore PIC32MX250F128B, che garantisce notevole memoria locale ed elevate velocità di elaborazione. I dati sono campionati a 125Hz. Oltre alla Cyton Board, che funge da modulo principale, vi è un modulo aggiuntivo chiamato *OpenBCI Daisy Module* che, attaccato sopra alla prima, permette di acquisire altri 8 canali per un totale di 16 canali, attraverso i quali si possono acquisire biopotenziali legati all’attività cerebrale (EEG), attività muscolare (EMG), e attività cardiaca (ECG). Il sistema comunica in modo wireless con il computer attraverso la *OpenBCI USB*. Esso può inoltre comunicare in modo wireless con ogni dispositivo e tablet compatibile con il protocollo Bluetooth Low Energy (BLE). [27]

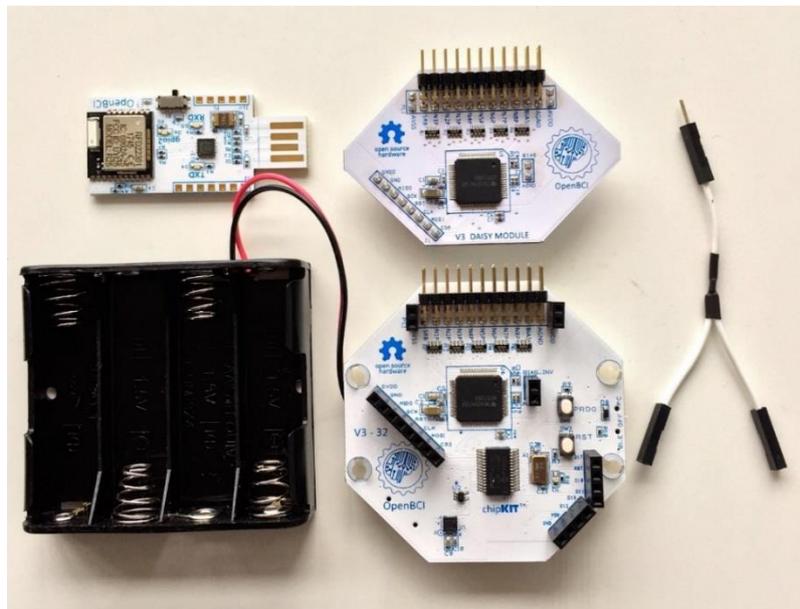


Figura 2.5. Componenti hardware del sistema OpenBCI utilizzato: si riconoscono la OpenBCI USB (in alto a sinistra), il modulo principale Cyton OpenBCI Board (in basso), e il modulo aggiuntivo OpenBCI Daisy Module (in alto). (tratta da [27])

Il vantaggio di questo sistema di acquisizione è quello di permettere una trasmissione dati efficace, veloce, real-time e soprattutto senza fili. Infatti, nel contesto operativo delle esperienze al CAVE, un sistema di acquisizione *wired* avrebbe reso difficoltoso il processo, in quanto avrebbe limitato notevolmente i movimenti dei volontari sul palco di prova. Pertanto, un sistema wireless come OpenBCI ha permesso ai soggetti una interazione realistica con lo scenario virtuale cui erano sottoposti e con i relativi oggetti in esso presenti, garantendo libertà di movimento e fruizione dell'intera area calpestabile del palco rialzato. Questo aspetto implicava però che il sistema Cyton + Daisy dovesse essere in qualche modo "attaccato" al soggetto. A tal proposito, nei periodi antecedenti l'inizio delle esperienze al CAVE, io insieme agli altri due tesisti abbiamo cercato la soluzione migliore per il riposizionamento degli elettrodi della cuffia, liberando la zona centrale per fare alloggiare la Board. In tale processo si è reso necessario "eliminare" dall'acquisizione l'elettrodo centrale Cz. La configurazione finale ottenuta è la seguente:

<i>Canale</i>	<i>Nome elettrodo</i>
1	F4
2	C4
3	T8
4	PO4
5	PO8
6	O2
7	F3
8	C3
9	T7
10	PO3
11	PO7
12	O1
13	ECG

Tabella 2.1. Elenco canali-elettrodi utilizzati con sistema OpenBCI. Notare l'ultimo canale, riservato all'ECG.

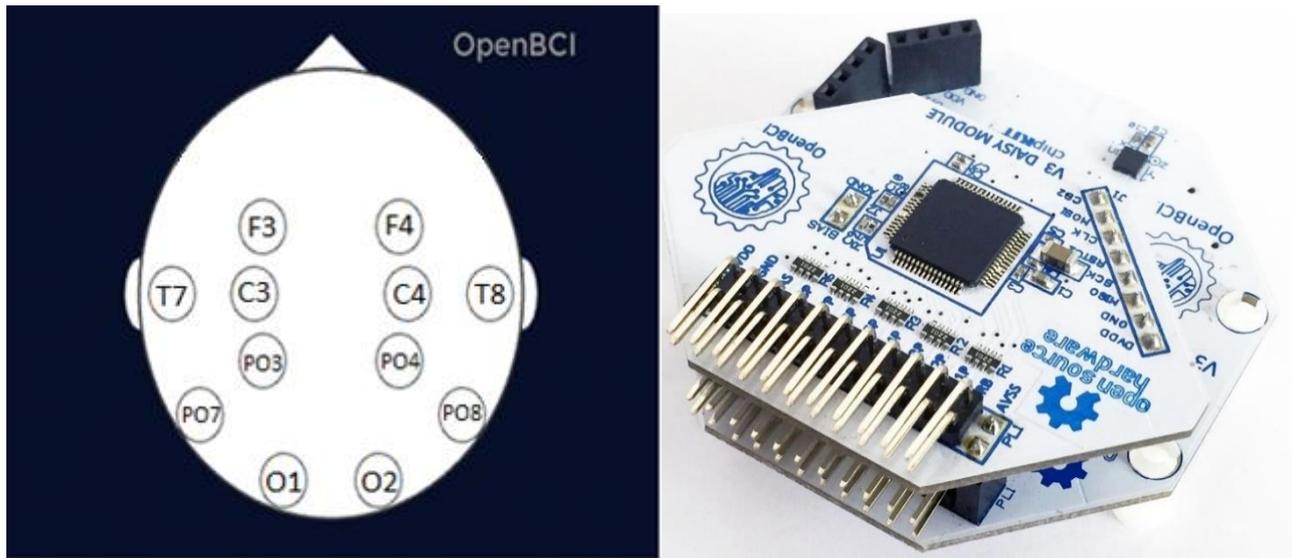


Figura 2.6. A sinistra la configurazione scelta per gli elettrodi sulla cuffia EEG. A destra il sistema Cyton+Daisy che abbiamo fissato sulla cuffia nell'area centrale compresa fra i sei elettrodi F3, F4, C3, C4, PO3, PO4. Nota: in aggiunta ai dodici canali EEG, il canale 13 è stato adibito ad ECG.

2.5 IL LABORATORIO DI INGEGNERIA BIOMEDICA (LIB)

Il Laboratorio di Ingegneria Biomedica (LIB), situato presso il Campus di Cesena (Università di Bologna), offre un punto di riferimento specializzato per la didattica e la ricerca nell'area biomedica. Il Laboratorio, che raccoglie in un unico spazio le diverse aree di interesse, offre venti postazioni di lavoro e un'ampia dotazione di strumentazioni biomediche. [28]

Nel LIB sono presenti numerose apparecchiature biomediche: macchina per dialisi, analizzatori di monitor di pressione invasivi, analizzatore di monitor di pressione non invasivo, analizzatori di sicurezza elettrica, elettrocardiografi, sistemi di acquisizione e analisi di segnali fisiologici, ventilatore polmonare alimentato da compressore, tester per ventilatori polmonari, analizzatore funzionalità respiratoria, manometri digitali. [28]

2.6 FINALITÀ DEGLI ESPERIMENTI IN LIB

Le acquisizioni sperimentali in LIB a Cesena consistono in una serie di esperimenti, precisamente quattro, concepiti per valutare, analizzando i tracciati EEG ed ECG, l'effetto di diverse componenti attentive (visiva, motoria, mentale), indotte nei soggetti durante lo svolgimento di quattro specifici task.

Nello specifico, sono state possibili un'analisi spettrale dei segnali EEG e di HR registrati durante le varie fasi di ciascun esperimento (ogni esperimento consiste di due fasi di relax, iniziale e finale, e di una fase centrale in cui il soggetto esegue uno specifico task), e un confronto fra i risultati dei quattro diversi esperimenti (gli esperimenti si distinguono per il diverso task svolto nella fase centrale, che coinvolge a seconda dell'esperimento, componenti attentive diverse).

2.7 FASI DEGLI ESPERIMENTI IN LIB

Complessivamente 30 volontari sani (diversi da quelli reclutati negli esperimenti del CAVE) hanno partecipato a questi esperimenti in LIB. Questi soggetti sono stati reclutati tra gli studenti dei Corsi di Laurea in Ingegneria Biomedica a Cesena (Università di Bologna). All'inizio di questo lavoro di tesi, la acquisizioni in LIB erano già state avviate da altri tesisti magistrali e io ho assistito a 4 acquisizioni per prendere atto della strumentazione/software utilizzati in fase di acquisizione, dei vari tipi di esperimenti (e task) e della varie fasi di ciascun esperimento.

A differenza di quanto accadeva per gli esperimenti presso il CAVE di Forlì, i soggetti volontari che si prestavano alle acquisizioni in laboratorio LIB a Cesena non necessitavano di alcuna fase di addestramento. Infatti, il contesto di queste esperienze era decisamente più semplice dal punto di vista operativo: non vi erano realtà virtuale, movement-tracking, megaschermi o palchi, ma solamente un volontario seduto ad una scrivania di fronte ad un monitor.

Invece, dal punto di vista delle tempistiche, qui si rendeva necessaria un'ora per completare l'intera acquisizione su ciascun soggetto che prevedeva infatti quattro

esperimenti ognuno della durata di 15 minuti (3 fasi da 5 minuti in ciascun esperimento).

Sebbene la strumentazione per l'acquisizione dei segnali EEG ed ECG ed il software utilizzato fossero diverse da quelle relative alle esperienze di Forlì, anche per quanto riguarda queste prove in laboratorio la prima fase da seguire era la "vestizione" del soggetto. La preparazione del volontario è identica a quella descritta al paragrafo 2.3: posizionamento della cuffia, iniezione di gel negli elettrodi EEG, fissaggio degli elettrodi ECG in corrispondenza delle scapole e scrub (pulitura e strofinamento) dei lobi auricolari per ospitare gli elettrodi di riferimento. L'unica differenza risiede nella maggiore accortezza e attenzione che l'operatore doveva dimostrare in questa specifica fase, dal momento che il sistema di acquisizione utilizzato (*Neurowave*, vedi seguito) non è wireless come OpenBCI, bensì basato su collegamenti a filo. Ciascun elettrodo sulla calotta aveva un filo di colore diverso che lo collegava al sistema di amplificazione, filtraggio, conversione ed elaborazione posizionato sul tavolo. L'utilizzo di un sistema di acquisizione non wireless rispecchia la natura operativa dell'esperimento: il soggetto volontario doveva rimanere seduto per tutta la durata della prova.



Figura 2.7. Uno dei soggetti volontari durante l'esperimento. Vista laterale della cuffia per EEG. (tratta da archivio personale)

Ogni soggetto veniva sottoposto a quattro esperimenti che verranno indicati con i termini *completo*, *mentale*, *lettura*, *movimento dito*; questa denominazione è basata sullo specifico task che ciascun esperimento prevedeva. Ogni soggetto veniva sottoposto a questi quattro esperimenti in ordine casuale. La suddivisione in fasi è la stessa per tutti e quattro gli esperimenti, che quindi si differenziano unicamente per il tipo di task implementato.

1. **Rilassamento iniziale:** il soggetto volontario vede davanti a sé uno schermo grigio con la scritta “*relax*” al centro, e per 5min deve rilassarsi e liberare la mente. Lo schermo è stato realizzato appositamente di un colore grigio poco luminoso per arginare il problema dell’affaticamento degli occhi dell’utente, lamentato dai primi volontari (inizialmente la schermata era bianca).
2. **Compito (task) centrale:** lo schermo mostra all’utente lo specifico task da eseguire. I quattro task, relativi ai quattro esperimenti, sono descritti di seguito. Ogni task da svolgere dura 5 minuti; il task altera lo stato di rilassamento precedente e comporta il coinvolgimento di componenti attentive e anche di stress (diverse a seconda del task da svolgere).

Task completo: al soggetto vengono presentate delle semplici operazioni di somma e sottrazione, ed è tenuto a rispondere, cliccando col mouse sugli appositi pulsanti, se il risultato dell’operazione è minore “<”, maggiore “>”, o uguale “=” ad un certo valore visualizzato a monitor. Questo task viene detto completo perché coinvolge tutte le seguenti componenti attentive: visiva (di lettura – non ad alta voce - dei caratteri alfanumerici a video), mentale (di computazione, per calcolare il risultato dell’operazione), motoria (di movimento del dito per rispondere esplicitamente con il click del mouse). Lo stato attentivo (e anche di stress del soggetto) è sollecitato richiedendo al soggetto (prima dell’inizio delle acquisizioni) di rispondere il più accuratamente ma anche il più velocemente possibile; per incentivare il soggetto a una risposta veloce, ogni volta che compare una nuova operazione a video, la schermata si ferma e non prosegue finché il soggetto non fornisce la risposta ma il trascorrere del tempo viene mostrato a video.

Task mentale: Anche in questo caso a video viene presentata una semplice operazione di somma e sottrazione, e al soggetto è richiesto se il risultato

dell'operazione è minore "<", maggiore ">", o uguale "=" ad un certo valore visualizzato a monitor. A differenza però del task completo, in questo task il soggetto si deve impegnare a pensare alla risposta corretta senza però premere alcun pulsante (e quindi senza fornire alcuna risposta esplicita). Ogni 5s viene presentata al soggetto una diversa operazione fino al termine dei 5 minuti. Questo task richiede il coinvolgimento di componenti attentive visive e mentali, ma sono escluse componenti motorie dato che il soggetto resta fermo per l'intera durata del task.

Task lettura: La presentazione a video è del tutto analoga a quella del task mentale, ovvero ogni 5s viene presentata al soggetto una diversa operazione (del tipo sopra descritta). In questo caso, però, il soggetto è tenuto a leggere (mentalmente, non ad alta voce) i numeri che vede sul monitor, in una modalità passiva, cioè senza svolgere l'operazione e pensare alla risposta corretta e senza premere alcun pulsante. In questo task sono dunque coinvolte solo componenti attentive di tipo visivo (lettura dei caratteri sul monitor), ma senza eseguire alcuna computazione né movimento.

Task movimento dito: In questo caso la schermata di volta in volta presentata all'utente non ha disequazioni, bensì solo i tre pulsanti di scelta. Si chiedeva al soggetto di premere un pulsante in modo casuale ogni 5-10s. In questo caso l'unica componente attivata è quella motoria.

Questi quattro task sono stati ideati per valutare l'effetto delle varie componenti attentive sui segnali EEG e di HR.

3. **Rilassamento finale:** il soggetto vede davanti a sé nuovamente lo schermo grigio di relax, e in questi ultimi 5min dell'esperimento deve rilassarsi e liberare la mente.

Anche in questo caso si riportano sinteticamente i risultati di questi esperimenti. Come fatto in precedenza si sottolinea che la valutazione di questi risultati è stata fatta da altri tesisti (della laurea magistrale) in altri lavori di tesi, essendo l'obiettivo di questa tesi limitato allo sviluppo di un'interfaccia grafica per la visualizzazione e l'analisi dei segnali EEG ed ECG.

- La potenza dei segnali EEG in banda alpha diminuisce durante la fase centrale di task rispetto alle fasi di rilassamento.
- La diminuzione è più evidente nel task completo rispetto agli altri task più semplificati; in particolari i cambiamenti minori di potenza in banda alpha si osservano nei task puramente visivo (task lettura) e nel task puramente motorio (task movimento dito)
- L'analisi spettrale dell'HR fornisce risultati coerenti con i risultati EEG. Il rapporto tra la potenza LF e la potenza HF (LF/HF) aumenta sistematicamente nelle fasi centrali di task rispetto alle fasi di relax. L'aumento maggiore del rapporto LF/HF si osserva durante il task completo, mentre i cambiamenti minori si hanno nel task lettura e nel task di movimento dito (puramente visivo e puramente motorio).

Questi risultati suggeriscono che l'attenzione modula significativamente sia la potenza in banda alpha dell'EEG che la variabilità della frequenza cardiaca e che il livello di questa modulazione cresce proporzionalmente all'impegno attentivo richiesto dal compito.

2.8 SETUP STRUMENTALE UTILIZZATO NEGLI ESPERIMENTI IN LIB: IL SISTEMA DI ACQUISIZIONE "NEUROWAVE"

Il set-up strumentale utilizzato per le acquisizioni in LIB si compone dei seguenti elementi: cuffia per EEG, elettrodi per EEG, elettrodi per ECG, amplificatore, desktop con software dedicato Neurowave e un notebook personale. Nel seguito si danno informazioni più dettagliate su alcuni di questi elementi, e brevi cenni su altri.

Tredici elettrodi in argento clorurato, con placche del diametro di 1 cm, sono tenuti fissi allo scalpo mediante una cuffia in materiale elastico, regolabile e adattabile di volta in volta secondo le misure di ogni singolo scalpo. Ogni elettrodo sulla cuffia presenta un foro centrale attraverso il quale, mediante una siringa con ago a punta smussa, viene inserito il gel conduttivo per assicurare una bassa impedenza di elettrodo. Essendo gli elettrodi fissati ad una calotta il loro posizionamento è

semplice, veloce e altamente riproducibile. Inoltre, il montaggio è abbastanza confortevole per il soggetto. [1]

La distribuzione degli elettrodi sulla cuffia EEG per questa prova è principalmente orientata verso la regione parieto-occipitale, sede delle onde *alpha* più evidenti. Inoltre, sono presenti alcuni elettrodi accessori: un elettrodo “a pinza” sul lobo dell’orecchio destro che funge da riferimento; un elettrodo centrale sulla fronte che funge da *ground*; infine, due elettrodi per la rilevazione dell’elettrocardiogramma (ECG) che vengono posti sulla cute del petto, in corrispondenza delle clavicole.

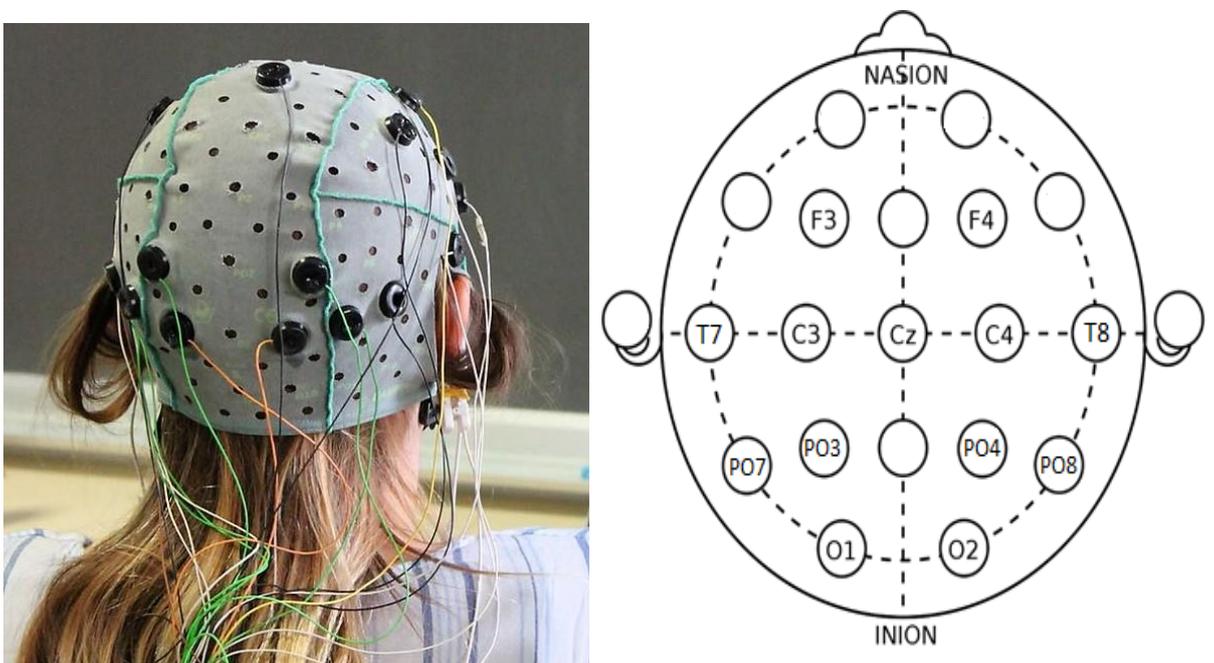


Figura 2.8. A sinistra una fotografia che ritrae uno dei soggetti volontari durante l’esecuzione della prova;(tratta da archivio personale). A destra una mappa della distribuzione degli elettrodi in questione; (tratta da [31]). Notare l’addensamento degli elettrodi in regione parieto-occipitale, sede dei ritmi *alpha* più evidenti.

<i>Canale</i>	<i>Nome elettrodo</i>
1	F4
2	C4
3	O2
4	T8
5	PO8
6	F3
7	C3
8	O1
9	T7
10	PO7
11	CZ
12	PO3
13	PO4
14	ECG

Tabella 2.2. Tabella canale-elettrodo relativa alle acquisizioni Neurowave in LIB, Cesena.

La parte di preelaborazione analogica è data dal sistema di amplificazione della *Braintronics* chiamato BRAINBOX® EEG-1166 Amplifier (vedi Figura 2.9).

Esso implementa al suo interno il circuito di controllo delle impedenze d'elettrodo.

Il brainbox è inoltre dotato di due masse attive G1 e G2 per sopprimere le principali interferenze (50/60Hz) in ambienti rumorosi. [30]



Figura 2.9. Sistema di amplificazione BRAINBOX® EEG-1166; (tratta da [30])

Il desktop che contiene il software Neurowave, consente di registrare, filtrare e visualizzare i segnali EEG estratti dal soggetto. È possibile un filtraggio con *filtro Notch a 50Hz* per eliminare il rumore di rete, non solo in visualizzazione bensì in fase di acquisizione: i dati che registra sono infatti già filtrati.

Il software Neurowave salva i dati ECG ed EEG di ogni soggetto in una cartella. Prelevando questa cartella e trasportandola sul proprio PC, si possono caricare in MATLAB® i dati che essa contiene e si può eseguire una analisi appropriata a questo esperimento.

Grazie al software Neurowave è possibile il controllo delle impedenze: vengono visualizzati, in tempo reale, i colori rosso (pessimo contatto), arancione (discreto contatto), e verde (ottimo contatto), per ogni elettrodo, in modo da comunicare all'operatore la necessità di inserire più gel conduttore nei fori degli elettrodi. Questa utilissima funzionalità non era invece presente in OpenBCI utilizzato negli esperimenti nell'ambiente CAVE.

Il notebook personale, posto a fianco del desktop Neurowave, era usato per lo svolgimento del task vero e proprio. È infatti sul video di questo notebook che si presentavano al soggetto le schermate per lo svolgimento dei diversi task.

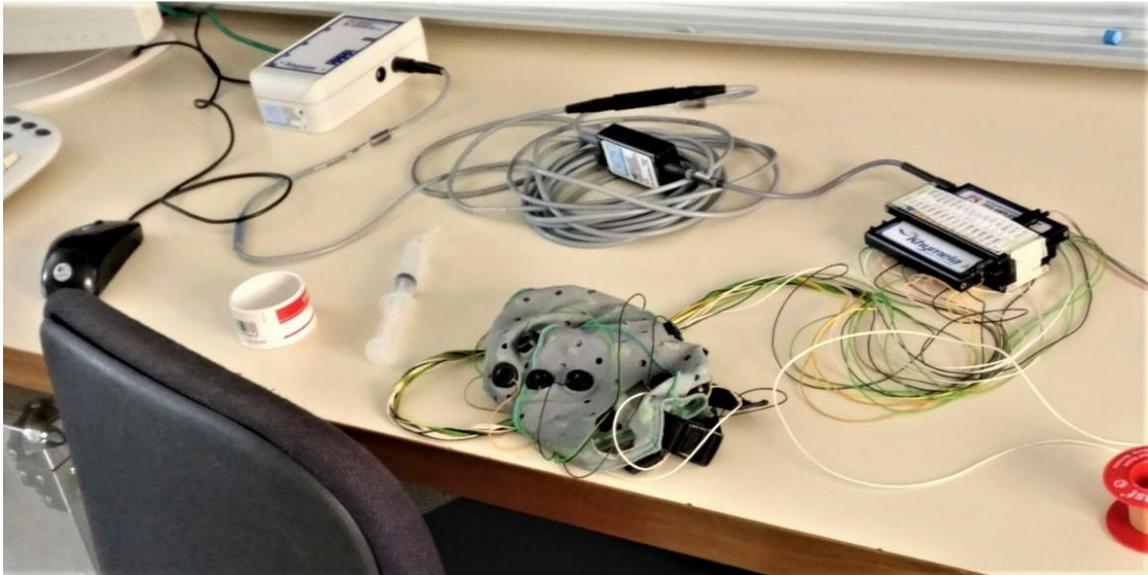


Figura 2.10. Fotografia che ritrae parte della strumentazione utilizzata per il suddetto esperimento. Si possono distinguere la cuffia per EEG, l'amplificatore della Braintronics, il cerotto adesivo per il fissaggio degli elettrodi ECG, e la siringa a becco smusso utilizzata per iniettare il gel conduttivo. (tratta da archivio personale)

2.9 CONSIDERAZIONI AGGIUNTIVE

In questa sezione si forniscono alcune precisazioni su come il segnale EEG, che come abbiamo visto origina dall'attività sinaptica dei neuroni nella corteccia cerebrale, si propaga sino alla superficie del cervello e viene 'captato' dai sensori (gli elettrodi), e sull'importanza di utilizzare un gel conduttivo, valide in generale per qualsiasi tipo di registrazione EEG di superficie, e nello specifico per entrambe le esperienze di laboratorio descritte nel presente capitolo. La conduzione di volume (*volume conduction*) è responsabile della propagazione del segnale EEG dentro il cervello. È possibile vedere la conduzione di volume come il processo tramite cui una moltitudine di ioni respinge gli ioni vicini con la stessa carica. Tali ioni respinti, a loro volta respingono ioni di ugual carica, producendo così un'onda di carica che viaggia attraverso lo spazio extracellulare. È questo il meccanismo basilare che permette al segnale di propagarsi entro il cervello. Il cervello, purtroppo, non ha un volume omogeneo: gli ioni non possono viaggiare attraverso nervi rivestiti di mielina o attraversare altre barriere. Le diverse densità dei tessuti possono impedire o permettere il flusso ionico, in base alle loro proprietà elettriche e fisiche. Per essere misurato con un elettrodo di superficie, il segnale elettrico deve viaggiare

dall'interno del cervello all'elettrodo, attraverso dura madre, cranio e scalpo. Una volta che il segnale raggiunge la frontiera dello scalpo, la conduzione di volume non può più avere luogo, poiché gli ioni non possono attraversarlo. Fra i volumi, una forma di capacità diviene responsabile della propagazione del segnale fino all'elettrodo. Si passa così da *volume conduction* a *capacitive conduction* (vedi Figura 2.11.). [9]

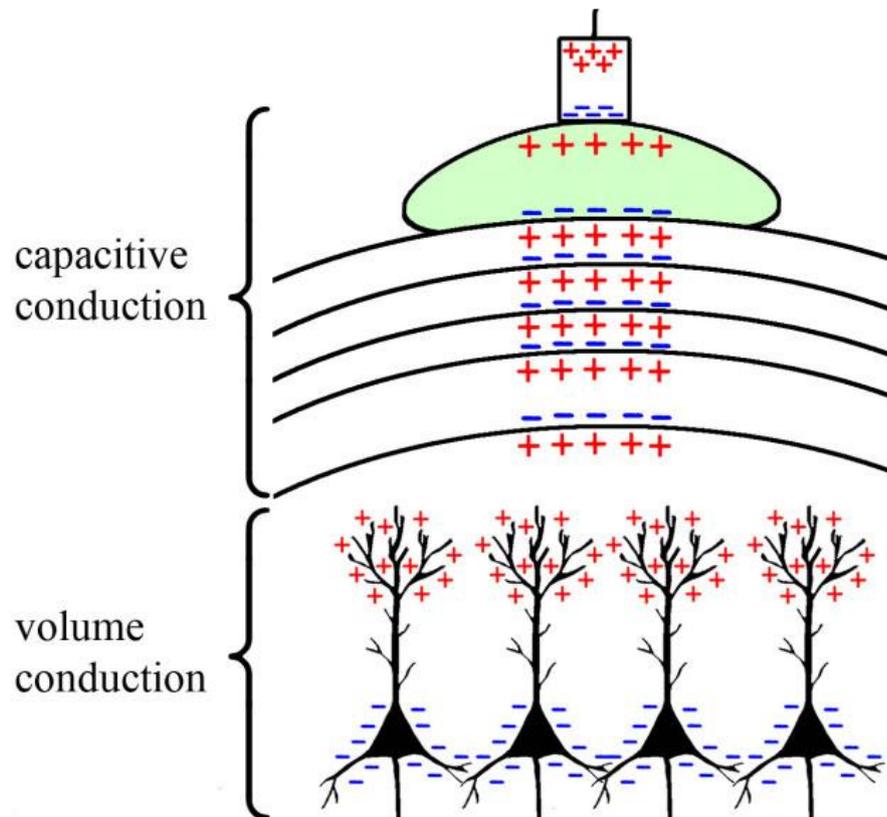


Fig. 2.11. Rappresentazione del meccanismo di conduzione del segnale EEG dal cervello all'elettrodo, attraverso volume conduction e capacitive conduction. Notare il gel conduttivo in verde. (tratta da [9])

Dati i numerosi strati di scarsi conduttori (cranio, cellule morte della pelle, capelli, aria negli interstizi), si rende fondamentale l'utilizzo di un particolare gel per elettrodi EEG. Idealmente questo gel altamente conduttivo dovrebbe saturare tutti gli spazi sottostanti l'elettrodo, riempiendo le bolle d'aria fra i capelli e quindi garantendo un percorso conduttivo dallo scalpo all'elettrodo [9]. Nella fattispecie, per le nostre esperienze di laboratorio è stato utilizzato *OneGel* della *EB Neuro*

(vedi Figura. 2.12), gel conduttore ad uso esterno completamente idrosolubile, ideale per acquisizioni EEG, ECG e EMG. Una pelle preparata correttamente garantisce una bassa impedenza ed una maggiore adesività dell'elettrodo sulla cute.

Per l'iniezione di tale gel nel foro centrale di ciascun elettrodo abbiamo utilizzato una siringa a becco smusso, mostrata in Figura 2.13.



Figure 2.12 e 2.13. Gel conduttore OneGel della EB Neuro e siringa a becco smusso; (tratte da archivio personale)

CAPITOLO 3

INTERFACCE GRAFICHE UTENTE: INTRODUZIONE A “GUIDE” E OGGETTI IN MATLAB

L'interfaccia grafica utente, parte fondante di questo lavoro, è stata implementata utilizzando il software MATLAB® 9.2 (Release 2017a). Prima di descrivere in dettaglio i passi, la realizzazione e le funzionalità della interfaccia utente sviluppate in questo lavoro per la visualizzazione e analisi di segnali EEG ed ECG (argomenti trattati nel successivo capitolo), si fornisce qui una panoramica sulle interfacce grafiche utente in MATLAB e sui metodi per la loro creazione.

3.1 LE INTERFACCE GRAFICHE UTENTE IN MATLAB

Le interfacce grafiche utente, note anche come GUI (dall'inglese *Graphical User Interface*), forniscono un controllo di tipo "punta e clicca" sulle applicazioni software (App), eliminando la necessità di imparare un linguaggio di programmazione o di digitare comandi per poter eseguire l'applicazione. MATLAB le implementa come oggetti di tipo *figure*.

Queste “figure” contengono i cosiddetti *uicontrol objects*, ovvero una serie di controlli intuitivi che permettono ad un utente di manipolare in modo interattivo una qualsiasi applicazione con l'utilizzo del mouse e della tastiera. Esempi di tali “oggetti” sono: tasti, campi editabili, barre di scorrimento e menu a scorrimento [32].

Dal punto di vista dell'utilizzatore finale, una GUI è il front-end di un'applicazione che automatizza un'attività o un calcolo. L'utente, infatti, spesso non è a conoscenza del livello sottostante (il livello del codice, noto al programmatore) che permette tali automatismi.

MATLAB offre due fondamentali vie per la creazione di una GUI: creazione interattiva e creazione programmatica. Su quest'ultima si daranno solo alcuni cenni, infatti è sulla creazione interattiva che si è basato questo progetto.

Creazione programmatica di una GUI MATLAB. Per avere un maggiore controllo e una maggiore flessibilità sulla progettazione e sullo sviluppo, è possibile creare un codice MATLAB che definisca le proprietà e i comportamenti di tutti i componenti dell'interfaccia grafica. MATLAB contiene una funzionalità integrata che aiuta a creare programmaticamente la GUI per la App desiderata, tramite librerie di funzioni specifiche. È possibile aggiungere finestre di dialogo, comandi di interfaccia utente (quali pulsanti e cursori) e contenitori (quali riquadri e gruppi di pulsanti) [32].

Creazione interattiva di una GUI MATLAB. Tramite l'ambiente di sviluppo GUIDE (vedi seguito), MATLAB fornisce gli strumenti per progettare interfacce utente destinate ad App personalizzate. Il Layout Editor di GUIDE consente di progettare graficamente la propria GUI. Successivamente, GUIDE genera automaticamente il codice MATLAB per costruire l'UI, che potrà essere modificato per programmare il comportamento di pulsanti, cursori, popup menu e altro [32].

3.2 L'AMBIENTE GUIDE DI MATLAB E IL LAYOUT EDITOR DI GUIDE

L'ambiente grafico di tipo *trascina-e-rilascia* ('drag-and-drop') messo a disposizione da MATLAB, chiamato *GUIDE (Graphical User Interface Design Environment)*, permette di ridurre notevolmente i tempi di sviluppo di un'interfaccia grafica, rispetto alla programmazione testuale (programmatica) rendendo il procedimento più intuitivo.

Per avviare GUIDE è sufficiente digitare "*guide*" sulla Command Window.

La finestra che si apre, chiamata GUIDE Quick Start, permette di selezionare una GUI vuota tra i modelli messi a disposizione oppure di aprire una GUI già esistente (vedi Figura 3.1).

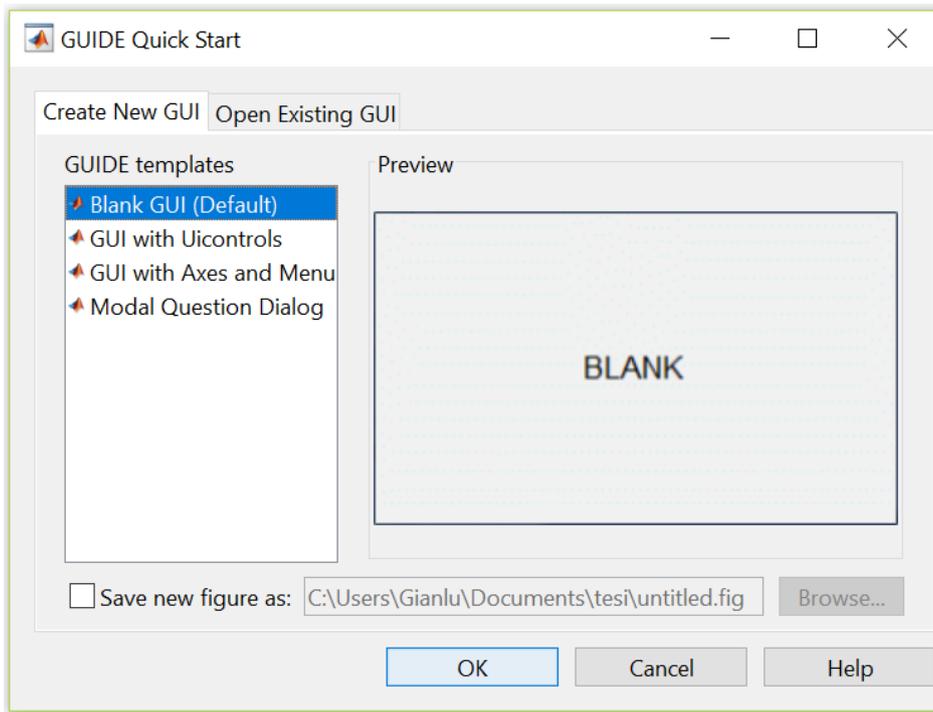


Figura 3.1. GUIDE Quick Start. È questa la finestra che viene visualizzata in seguito alla digitazione della parola “guide” sulla Command Window di Matlab; (tratta da MATLAB attraverso il tasto Stamp).

Per creare una nuova GUI si seleziona la prima voce “Blank GUI (Default)”. In questo modo apparirà il Layout Editor mostrato in Figura 3.2.

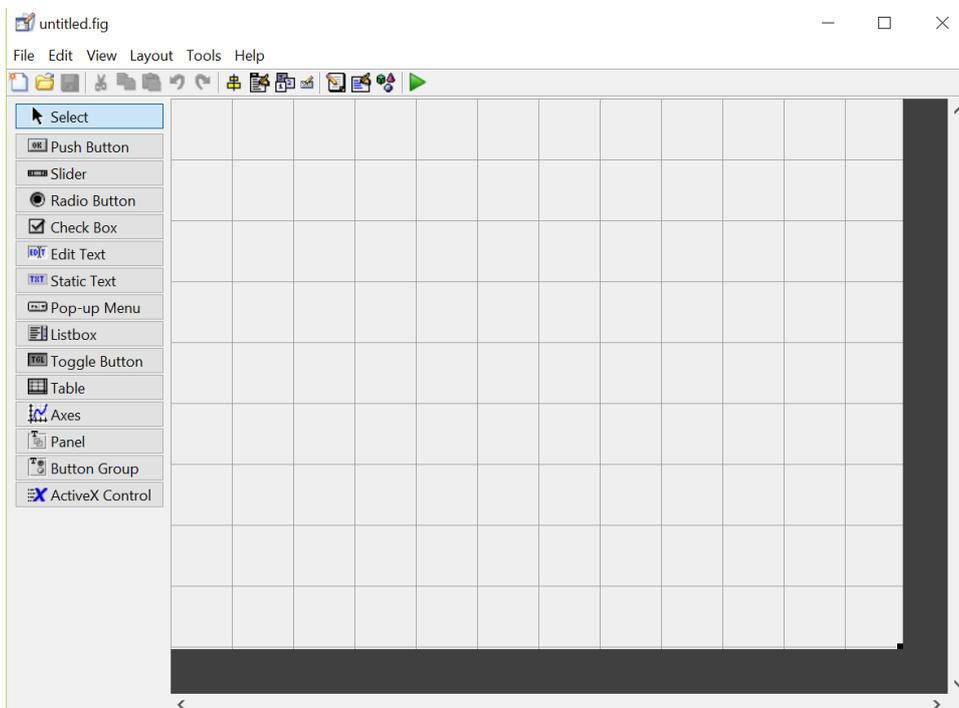


Figura 3.2. Layout Editor di GUIDE; (tratta da MATLAB attraverso il tasto Stamp).

L'Editor è così organizzato: presenta in alto una barra degli strumenti, a sinistra la *palette* degli oggetti che è possibile inserire nella propria interfaccia e al centro l'area di lavoro (parte suddivisa in quadretti).

L'area di lavoro ha dimensioni regolabili tramite mouse trascinando i vertici del rettangolo.

L'aspetto grafico della GUI associata alla propria App è la prima cosa da progettare, scegliendo quali degli oggetti disponibili si vogliono inserire, e definendone disposizione e dimensioni.

È bene precisare che l'aspetto grafico impostato in questa fase non è definitivo ed imm modificabile (si tratta solo di un primo step). Infatti, il punto di forza di questo ambiente di sviluppo è che permette modificazioni successive, ad esempio per arricchire in tempi diversi la App di nuove funzionalità allo scopo di soddisfare nuove esigenze.

3.3 GLI OGGETTI

Nella *palette* sulla sinistra dell'Editor sono presenti i già citati *uicontrol objects* (vedi Figura 3.3), ossia gli oggetti messi a disposizione del programmatore dal GUIDE di Matlab per popolare la propria interfaccia grafica utente.

Per aggiungere uno dei suddetti oggetti all'interfaccia è sufficiente fare clic prima su di esso, quindi sulla zona in cui lo si vuole posizionare; in alternativa, è possibile trascinarlo e rilasciarlo nell'area desiderata.

Essi possono essere controlli (pushbutton, slider, edit text), elementi statici (panel, static text), menu e oggetti utilizzati per visualizzare dati grafici.

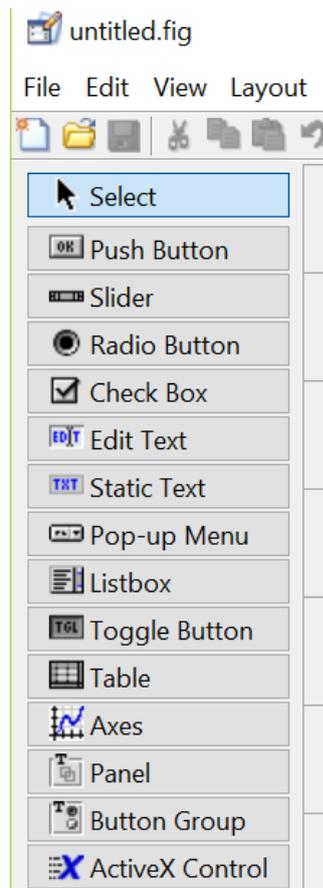


Figura 3.3. Palette degli uicontrol objects, gli oggetti messi a disposizione del programmatore dall'ambiente GUIDE di Matlab per popolare la propria interfaccia grafica utente; (tratta da MATLAB attraverso il tasto Stamp).

Di seguito vengono elencati e descritti tutti gli *objects* nella palette:

Push Button: pulsante che richiama un evento con il clic del mouse su di esso.

Slider: cursore a scorrimento per l'incremento o decremento del valore di una variabile; è possibile impostare il passo di avanzamento discreto del cursore.

Radio Button: pulsante circolare per l'attivazione di una funzionalità; i suoi possibili stati sono cerchietto vuoto e cerchietto pieno. Si attiva e disattiva al clic del mouse su di esso.

Check Box: riquadro di selezione per l'attivazione di una funzionalità, agisce come il Radio Button. I suoi possibili stati sono quadratino vuoto e quadratino spuntato.

Edit text: campo che consente all'utente di interagire con un oggetto testuale, editando un testo (stringa di lettere, numeri o caratteri).

Static text: mostra una stringa di testo in una casella; il testo viene scelto dal programmatore e non può essere modificato dall'utente.

Pop-up menu: menu a tendina che fornisce una lista di opzioni mutuamente esclusive. È l'utente a selezionare quella di suo interesse tramite un clic. Terminata la scelta la tendina del menu si minimizza.

Listbox: analogamente al precedente, mostra una lista di opzioni selezionabili (vengono mostrate tutte, senza tendina). È presente un cursore a lato per scorrere le diverse opzioni.

Toggle Button: pulsante con funzionamento simile al Push Button, con la differenza che il pulsante, una volta premuto, permane in tale stato fino al successivo clic su di esso; in questo modo implementa due funzionalità in base al proprio stato (premutato o rilasciato).

Table: oggetto che crea una tabella a doppia entrata (impostabile dal programmatore); permette la selezione di uno degli elementi della matrice generata.

Axes: oggetto che permette di visualizzare grafici nella GUI. Una volta trascinato sull'interfaccia, esso si presenta come un rettangolo bianco. Le sue dimensioni, regolabili tramite gli angoli, indicano le dimensioni del *plot* che ospiterà.

Panel: area in cui sono raggruppati più controlli. È principalmente usata per suddividere in sezioni l'intera area dell'interfaccia. Funge da contenitore all'interno del quale raccogliere diversi oggetti tra quelli sopra e sotto elencati.

Button Group: è un contenitore che ha la funzione di gestire la selezione mutuamente esclusiva di Radio Button e Toggle Button.

ActiveX Control: permette al programmatore di scegliere per la sua GUI uno dei componenti interattivi presenti nel Sistema utilizzato, caratterizzati dall'intercomunicazione fra l'applicativo MATLAB e i programmi del Sistema Operativo. Oltre ai pacchetti di default offerti da Microsoft Windows® (cornici, immagini, barre di scorrimento e simili), ulteriori plug-in sono possibili in base ai diversi programmi installati nello specifico laptop. Ad esempio, se è installato

Windows Media Player si avrà a disposizione un *object* di visualizzazione multimedia dentro alla GUI, completo dei consueti pulsanti “play”, “stop”, “volume” e “tutto schermo”; se è installato il pacchetto *Office* è disponibile un oggetto calendario; se è installato *Acrobat Reader* di Adobe® sarà presente un oggetto per la lettura dei PDF internamente alla GUI; infine, se è installato *LabVIEW* di National Instruments® saranno disponibili oggetti per visualizzare risultati, gestire le *query* e ricercare i dati sorgente.

3.4 LE PROPRIETÀ DEGLI OGGETTI

Ogni oggetto di una GUI possiede un certo numero di proprietà. Queste possono essere esaminate e modificate mentre si realizza la parte grafica, grazie allo strumento offerto da GUIDE, chiamato *Property Inspector*, oppure possono essere modificate ed elaborate in fase di scrittura del codice nel file .m, attraverso le funzioni “get” e “set”, descritte nel seguito. Modificando le proprietà degli oggetti, il programmatore ha la possibilità di personalizzare notevolmente aspetto e funzionalità della propria interfaccia, in modo tale da rispondere ad ogni sua esigenza.

Il *Property Inspector (PI)*, accessibile cliccando l'apposita icona  del Layout Editor di GUIDE oppure digitando “inspect” sul prompt dei comandi, permette al programmatore di modificare interattivamente gli oggetti, come gli oggetti grafici. Quando si seleziona uno di tali oggetti, il PI mostra una lista delle sue proprietà, i cui valori possono essere modificati editando i campi che appaiono a fianco a ciascuno dei nomi.

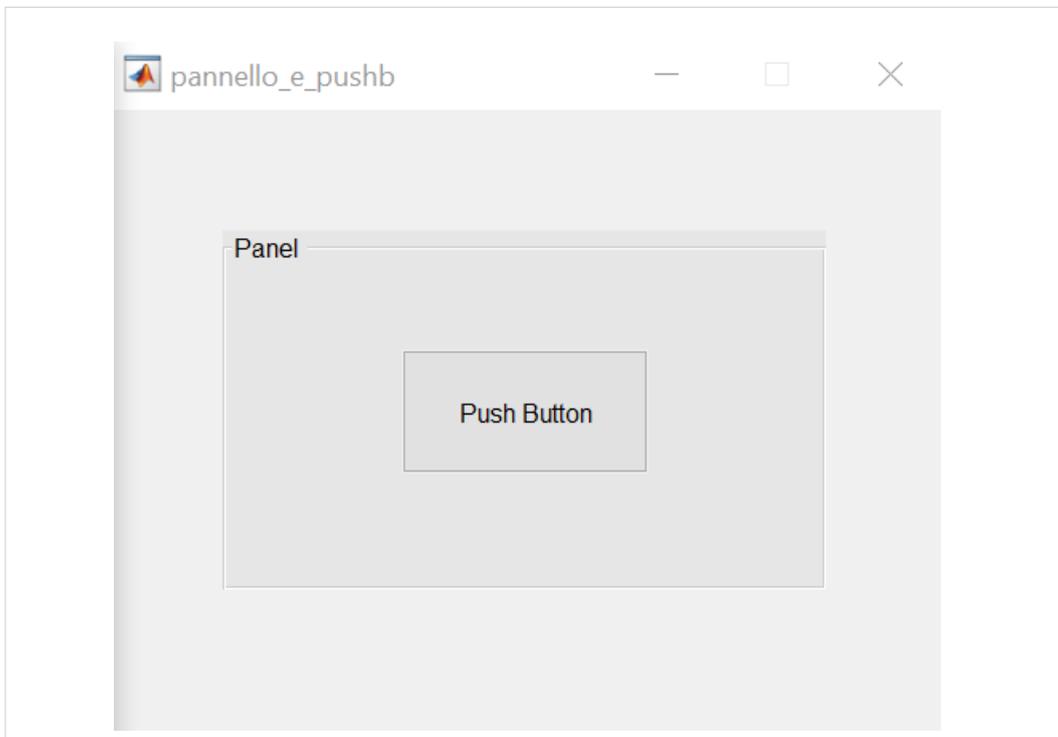
La finestra del PI è organizzata in sottocategorie, che dividono le proprietà dell'oggetto in base al contesto a cui si riferiscono, ad esempio aspetto grafico (*Appearance*), stile scrittura (*Font style*), posizione e dimensioni (*Position and Size*), e nomi di identificazione (*Identifiers*).

Dato un oggetto *panel*, è possibile, ad esempio, impostare il colore del suo sfondo, lo spessore dei suoi bordi ed il colore della sua ombreggiatura. Dato un oggetto

pushbutton, invece, sono modificabili, fra le altre proprietà, la dimensione, il font e lo stile della scritta che compare al suo interno.

Tra le proprietà che caratterizzano un oggetto, per il programmatore è fondamentale il “Tag”. Esso è il nome univoco che assume l’oggetto all’interno del codice; tale nome viene utilizzato per definirne funzioni e caratteristiche. Di default, MATLAB utilizza il nome del tipo di oggetto seguito da un numero sequenziale, ma è consigliabile, soprattutto se si deve creare un’interfaccia con molti oggetti dello stesso tipo, cambiare i Tag di default con nomi che facciano riferimento alle funzioni che svolgono. Ad esempio, chiameremo *compute_HR_button* e *notch_filter_EEG_button* i Tag rispettivamente del pulsante che calcola la frequenza cardiaca dal segnale ECG e del pulsante che filtra il segnale EEG con filtro Notch.

Per aiutare la comprensione del concetto di proprietà degli oggetti grafici, si dà l’esempio di una semplice GUI costituita essenzialmente da due oggetti: un *panel* ed un *pushbutton*. In Figura 3.4 (che continua a pagina seguente) sono mostrate la GUI, che si presenta all’utente, e la lista delle proprietà, dei due oggetti (pannello e pulsante), così come appaiono di default,



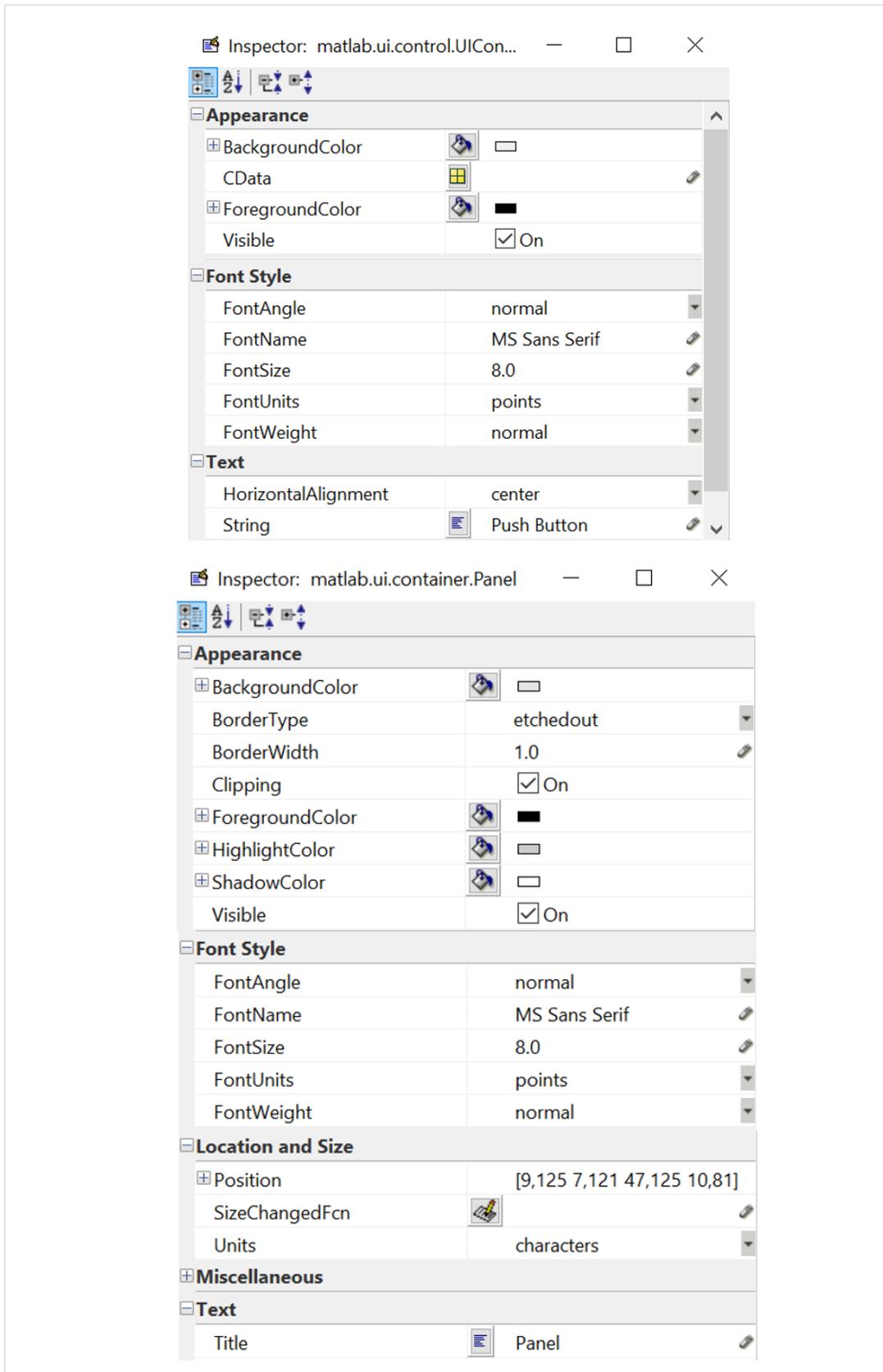
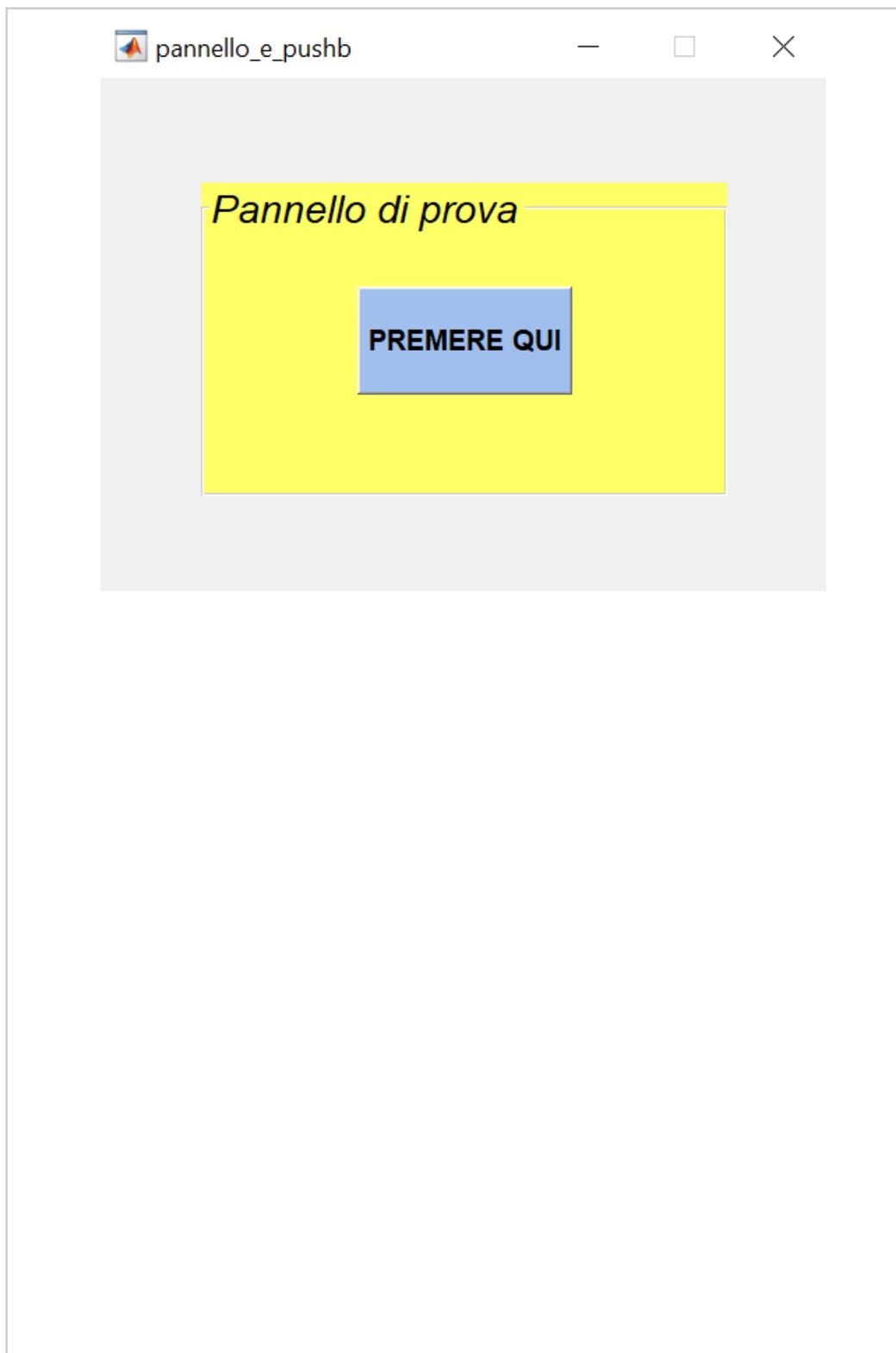


Figura 3.4 Una semplice GUI e le proprietà di default dei rispettivi oggetti ivi contenuti; (tratta da MATLAB attraverso il tasto Stamp).

Dopo aver illustrato come si presenta la GUI caratterizzata dalle proprietà di default, ho modificato, a titolo d'esempio, alcune di esse.

Il risultato è illustrato in Figura 3.5 (che continua a pagina seguente).



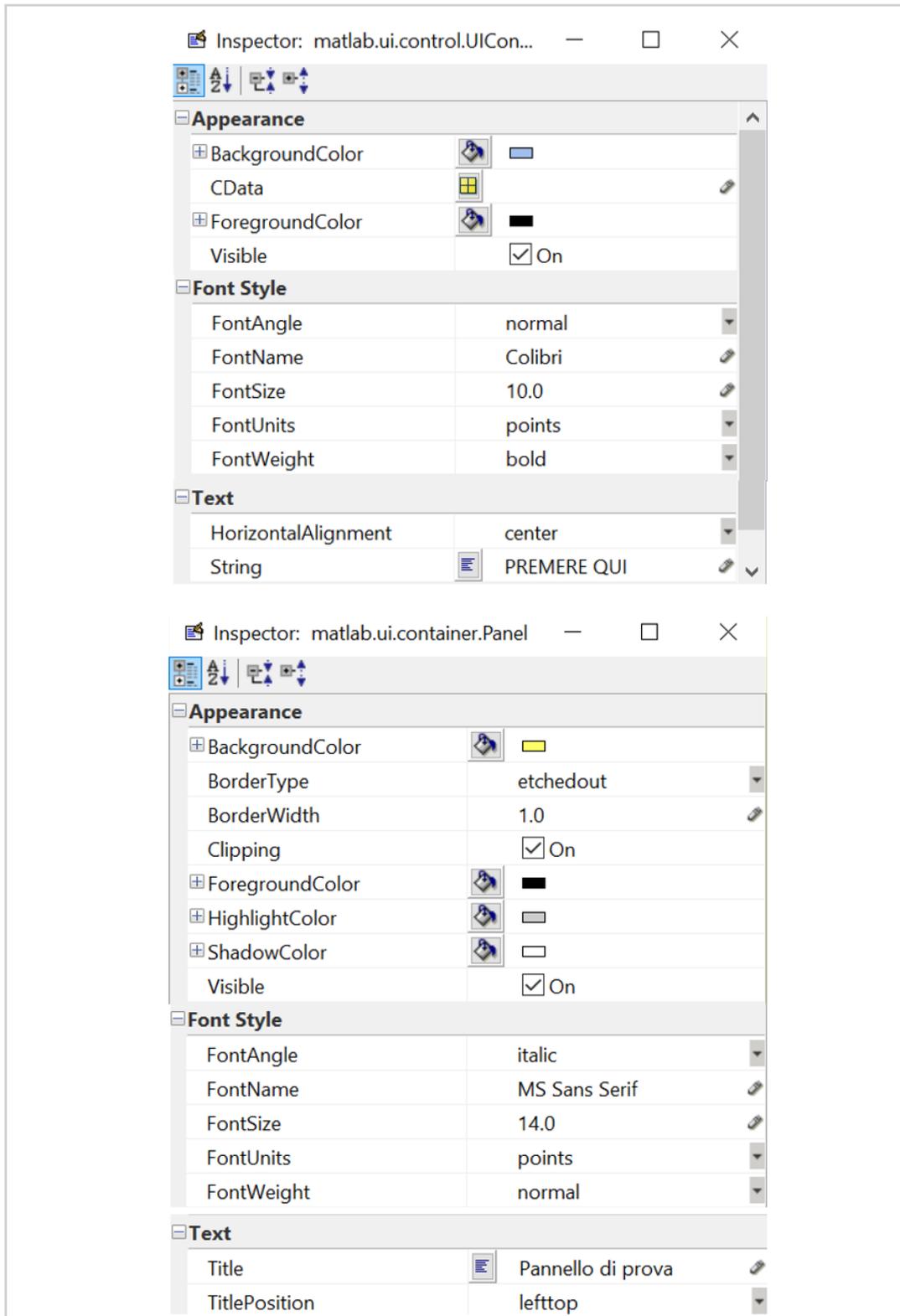


Figura 3.5. Una semplice GUI e le proprietà modificate dei suoi oggetti; (tratta da MATLAB attraverso il tasto Stamp).

Come si evince confrontando le precedenti due figure, per quanto riguarda il pushbutton sono stati modificati i campi `BackgroundColor` (per impostare lo sfondo azzurro), `FontName` (per impostare lo stile di font Colibri), `FontSize` (per cambiare la dimensione dei caratteri da 8 a 10 punti), `FontWeight` (per impostare lo stile grassetto) ed infine il campo `String` (per visualizzare la scritta PREMERE QUI sul pulsante). Del pannello invece è stato cambiato il colore di sfondo (ora giallo) ed è stato impostato un nuovo titolo, con dimensione 14 punti e stile (`FontAngle`) corsivo.

Come precedentemente anticipato, un programmatore può decidere di modificare le proprietà di un oggetto senza servirsi delle funzionalità messe a disposizione dall'ambiente grafico del Property Inspector, bensì tramite le due funzioni *get* e *set*, direttamente all'interno del codice Matlab nel file `.m` generato automaticamente in associazione alla GUI creata (vedi sotto-capitolo 3.5).

La prima di queste funzioni, *get*, permette di acquisire il valore di una proprietà. Ad esempio, l'istruzione:

```
get(handles.inseriscicolore, 'String' )
```

permette di leggere quale insieme di caratteri caratterizza la proprietà “*String*” dell'oggetto individuato dal Tag “*inseriscicolore*” (un edit text).

La seconda di queste funzioni, *set*, permette di impostare il valore di una proprietà. Ad esempio, l'istruzione:

```
set(handles.uipanel, 'BackgroundColor', 'green')
```

permette di impostare il colore verde nella proprietà “*BackgroundColor*” dell'oggetto individuato dal Tag “*uipanel*”.

Per identificare l'oggetto dobbiamo utilizzare la struttura *handles*, che contiene i puntatori agli oggetti presenti all'interno dell'interfaccia.

3.5 GENERAZIONE DEL CODICE

Ogni volta che si modifica l'interfaccia grafica associata alla propria App attraverso gli strumenti del Layout Editor e si salvano le modifiche (facendo clic sull'icona di salvataggio sulla barra degli strumenti del Layout Editor), l'interfaccia realizzata viene appunto salvata e MATLAB genera automaticamente due file con lo stesso nome (il nome dato alla GUI) ma con diversa estensione.

Infatti, un file *.fig* contiene la realizzazione grafica della GUI, ed un file *.m* contiene il codice Matlab generato automaticamente associato alle proprietà grafiche impostate tramite Layout Editor. Questo file *.m* può essere poi modificato dal programmatore per inserire i codici di controllo relative alle funzioni specifiche (chiamate *Callback*) da associare ai vari oggetti presenti nella GUI (ad es., se si inserisce nella GUI un push button 'Plot EEG' per la visualizzazione dei segnali EEG, all'interno del file *.m* sarà necessario associare a tale oggetto il codice che carica i segnali EEG nell'workspace di Matlab e implementa la visualizzazione di tali segnali EEG in funzione del tempo).

Uno degli aspetti caratteristici di questo modo di programmare una GUI è infatti che, tramite GUIDE, una parte del codice contenuto nel file *.m* è automaticamente generato da MATLAB. Essa comprende un *codice di inizializzazione* dell'interfaccia stessa (che non deve essere modificato), la *OpeningFcn*, che viene eseguita un istante prima che l'interfaccia si renda visibile all'utente (è in questo spazio che generalmente si inizializzano le variabili), la *OutputFcn*, le cui uscite sono restituite sulla Command line alla chiusura dell'interfaccia, ed infine le funzioni *Callback* relative ai singoli *objects* presenti nella GUI.

Spetta al programmatore definire entro le funzioni Callback la sequenza di comandi che deve essere eseguita nel momento in cui l'oggetto è attivato. Infatti, quando l'utente preme un pulsante o inserisce un testo da tastiera si attiva la Callback corrispondente.

Si dia l'esempio di una semplice interfaccia grafica, che ho chiamato *codificacolore*, caratterizzata da quattro oggetti: due static text, un pushbutton ed un edit text.

La presente GUI (vedi Figura 3.6) chiede all'utente di fare clic con il mouse sul pulsante pushbutton. Tale azione genera un vettore di tre numeri casuali compresi fra 0 ed 1, arrotondati alla seconda cifra decimale, che rappresentano la codifica RGB normalizzata di un colore.

Tale colore viene impostato come sfondo del pulsante stesso, e i tre numeri della codifica vengono stampati a video all'interno dell'edit box.



Figura 3.6. GUI “codificacolore”; (tratta da MATLAB attraverso il tasto Stamp)

All'interno del file *codificacolore.m* generato da Matlab vi sono le funzioni di inizializzazione, apertura, Output e Callback precedentemente citati.

A titolo d'esempio si riportano in Figura 3.7 il codice di inizializzazione della GUI, in Figura 3.8 le sue funzioni *OpeningFcn* e *OutputFcn*, ed in Figura 3.9 il codice di controllo del pushbutton.

```

function varargout = codificacolare(varargin)
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @codificacolare_OpeningFcn, ...
                  'gui_OutputFcn',  @codificacolare_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

```

Figura 3.7. Codice di inizializzazione della GUI “codificacolare”; (tratta da MATLAB attraverso il tasto Stamp)

```

% --- Executes just before codificacolare is made visible.
function codificacolare_OpeningFcn(hObject, eventdata, handles, varargin)
% Choose default command line output for codificacolare
handles.output = hObject;
% Update handles structure
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = codificacolare_OutputFcn(hObject, eventdata, handles)
% Get default command line output from handles structure
varargout{1} = handles.output;

```

Figura 3.8. Codici di apertura ed output della GUI “codificacolare”; (tratta da MATLAB attraverso il tasto Stamp)

```

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
colore=round(rand(1,3),2); %RGB
set(handles.pushbutton1, 'BackgroundColor', colore)
set(handles.Redit1, 'String', (num2str(colore)))

```

Figura 3.9. Codice di controllo, all’interno della funzione di Callback, del pushbutton1; (tratta da MATLAB attraverso il tasto Stamp)

Si tenga presente che al *Push Button* della GUI “codificacolore” è stato assegnato il Tag “pushbutton1”, mentre all’Edit Text è stato assegnato il Tag “Redit1”.

Come mostrato nel codice di Figura 3.9, la prima azione che il programma esegue dopo che l’utente ha premuto il pushbutton è quella di generare un vettore numerico chiamato *colore*.

Esso è costituito da tre numeri casuali compresi fra 0 ed 1 approssimati alla seconda cifra decimale. La generazione del vettore di numeri casuali è ottenuta tramite il comando *rand(n,m)* che, più precisamente, genera una matrice di n righe ed m colonne di numeri casuali nell’intervallo 0-1. Pertanto, utilizzando *rand(1,3)* si genera una matrice di una riga e tre colonne, ossia un vettore di tre numeri. L’arrotondamento a due cifre decimali è invece ottenuto tramite il comando *round(...,2)*, in cui “2” indica appunto le cifre decimali scelte.

Una volta generato il vettore *colore*, il codice continua con due istruzioni di *set*. La sintassi di tali funzioni, come già visto precedentemente, prevede un primo argomento che rappresenta il puntatore (contenuto in un campo della struttura *handles*) all’oggetto di cui voglio modificare una proprietà, un secondo argomento che indica la proprietà da modificare, ed un terzo argomento che rappresenta il nuovo valore da attribuire a tale proprietà.

Nello specifico, la prima istruzione di *set* imposta lo sfondo del *pushbutton1* al colore codificato nel vettore *colore*; mentre la seconda istruzione di *set* assegna il vettore *colore* all’interno del campo “String” dell’edit box *Redit1*, per fare questo però occorre convertire tale vettore da vettore numerico a stringa di caratteri, e ciò è possibile grazie al comando *num2str()*.

Per concludere la trattazione dell’esempio, si mostrano in Figura 3.10 alcuni esempi di funzionamento di questa semplice interfaccia grafica. Notare che, a clic successivi, cambia il colore del pulsante e contestualmente viene aggiornato il vettore RGB per la codifica dello stesso.

Così, al primo clic il pulsante diviene viola (0.65 0.41 0.82), al secondo clic diviene azzurro (0.22 0.67 0.84), poi marrone (0.76 0.38 0.22) ed infine verde limone (0.64 0.96 0.24).



Figura 3.10. Esempi di funzionamento della GUI "codificacolore"; (tratta da MATLAB attraverso il tasto Stamp)

CAPITOLO 4

ORGANIZZAZIONE E FUNZIONALITÀ DELL' INTERFACCIA GRAFICA UTENTE REALIZZATA

La GUI è stata realizzata con lo scopo di consentire un'appropriata visualizzazione dei segnali acquisiti negli esperimenti descritti nei capitoli precedenti, e quindi di condurre un'analisi spettrale dei segnali.

La GUI dunque rende possibile investigare la presenza di correlazioni tra le caratteristiche spettrali dei segnali e lo stato emotivo/attentivo del soggetto modulato durante l'esperimento.

In particolare, poiché ciascuno dei due esperimenti ideati prevedeva fasi di assenza di compito e di stimolazione alternate a fasi di stimolazione ed esecuzione di compiti, la GUI si è concentrata sulla computazione e visualizzazione delle caratteristiche spettrali in queste diverse fasi per evidenziarne, graficamente e quantitativamente, eventuali differenze.

Si sottolinea tuttavia che l'analisi di tali differenze non è stato l'oggetto di questo lavoro, il quale invece si è concentrato sul realizzare, attraverso la GUI, tools di facile utilizzo per consentire tale analisi.

Nella realizzazione della presente GUI, mi sono servito dell'ambiente GUIDE, descritto nel capitolo precedente. Pertanto, questo è un esempio di creazione di una GUI per via interattiva.

L'aspetto finale dell'interfaccia realizzata è mostrato in Figura 4.1. Nel seguito si daranno spiegazioni sui passi che l'utilizzatore deve seguire per fruire della GUI, e di come sono state implementate le diverse funzionalità di cui essa dispone.

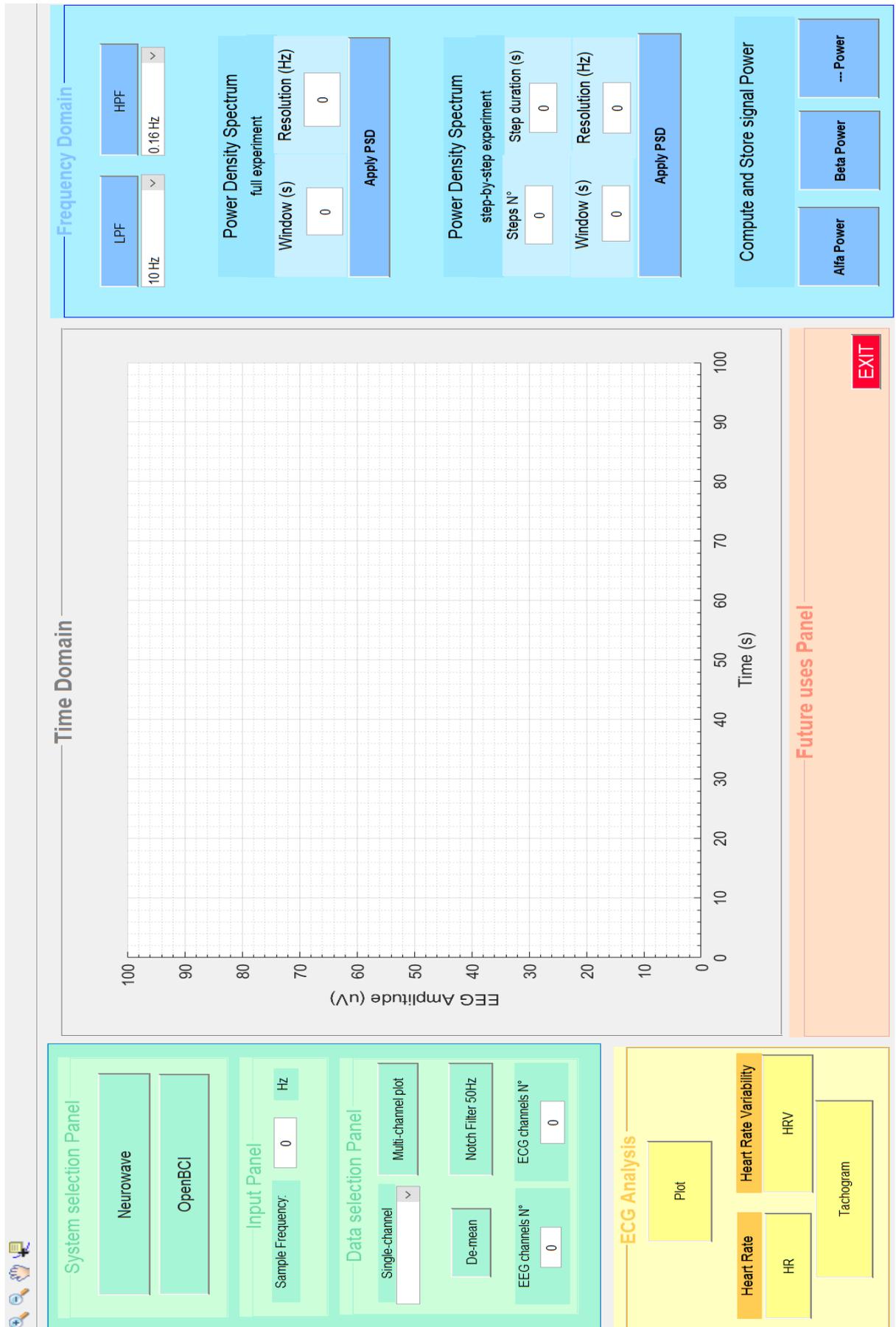


Figura 4.1. Visione d'insieme della Graphical User Interface realizzata; (tratta da Matlab attraverso il tasto Stamp)

4.1 STRUTTURA E SUDDIVISIONE DELLA GUI

L'interfaccia è stata organizzata in 4 macro-aree funzionalmente distinte, facilmente riconoscibili tramite le diverse colorazioni:

- Selezione ed Inserimento (area di colore verde);
- Visualizzazione EEG nel dominio dei tempi (area di colore grigio);
- Analisi EEG nel dominio delle frequenze (area di colore blu);
- Analisi ECG (area di colore giallo);

L'area **Selezione ed Inserimento** è la prima a cui l'utente accede. Essa è costituita da tre sotto-pannelli: *System selection Panel*, *Input Panel* e *Data selection Panel*. Il primo serve per selezionare il sistema di acquisizione utilizzato nell'esperimento (*Neurowave* od *OpenBCI*); il secondo ha lo scopo di permettere all'utente di inserire da tastiera, all'interno dell'edit-box dedicato, il valore della frequenza di campionamento (*Sample Frequency*) con cui sono stati acquisiti i segnali dal sistema in questione; infine, il terzo permette all'utente di scegliere se visualizzare il grafico relativo ai dati EEG del singolo elettrodo (*Single-channel*), oppure se visualizzare in parallelo i segnali EEG di tutti gli elettrodi acquisiti (*Multi-channel plot*). Le rimanenti funzionalità del *Data selection Panel* verranno illustrate nel seguito.

L'area colorata in grigio è dedicata alla visualizzazione dei segnali **EEG nel dominio dei tempi**. In base alla scelta dell'utente nel precedente pannello, sono possibili il grafico relativo al singolo elettrodo o quello di tutti gli elettrodi insieme. L'asse delle ascisse rappresenta la scala dei tempi in secondi (s), mentre su quello delle ordinate vi è l'ampiezza del segnale EEG in microvolt (μV). I dati in ingresso alla GUI, provenienti dal sistema di acquisizione scelto, non sono originariamente in 'secondi', bensì in 'campioni' (*samples*). Detto N il numero totale di campioni acquisiti nel segnale EEG ($N=length(data)$), per ottenere il vettore dei tempi è sufficiente dividere il vettore di lunghezza N per la frequenza di campionamento F_s :

$$tempi = [0: 1: N - 1]/F_s$$

Si noti che nell'equazione sopra, si è scelto di assegnare il primo campione al tempo $t = 0$.

L'area blu è dedicata invece ad operazioni ed analisi **EEG nel dominio delle frequenze**. Infatti, essa fornisce la possibilità all'utente di impostare le frequenze di taglio dei filtri passa-basso e passa-alto per operare un filtraggio in frequenza del segnale, al fine di eliminare disturbi alle alte frequenze e trascurare contributi poco utili dal punto di vista informativo alle basse frequenze. Questo stesso pannello contiene inoltre i pulsanti adibiti all'analisi spettrale, ed in particolare al calcolo della densità spettrale di potenza (PSD) fornendo la possibilità di calcolare la PSD sia sull'intera durata del segnale, sia su sezioni del segnale corrispondenti alle varie fasi dell'esperimento. Infine, vi è la possibilità di calcolare la potenza del segnale in banda *alpha* e *beta* tramite gli appositi pushbutton, e salvare i risultati ottenuti.

La sezione di colore giallo identifica invece l'area funzionalmente dedicata all'**analisi elettrocardiografica**. Il pushbutton "Plot" permette di visualizzare il segnale ECG ricavato dai due elettrodi posti sulle clavicole in un grafico tempo-ampiezza in una nuova figura, separata dalla GUI. Una volta visualizzato nei tempi tale segnale, è possibile, tramite gli appositi pulsanti, calcolare e visualizzare la frequenza cardiaca (*Heart Rate, HR*) e condurre un'analisi sulla sua variabilità (*Heart Rate Variability, HRV*), computando la densità spettrale di potenza, la potenza del segnale nelle bande LF ed HF, ed anche il rapporto LF/HF.

La Figura 4.1 rivela che è presente in realtà anche una quinta sezione, chiamata **Future uses Panel**, che, come suggerisce il nome, è stata progettata per usi futuri. Infatti, inserito ancora vuoto, tale pannello permette modifiche ed aggiunte alla GUI qualora nascessero esigenze nuove. La GUI può essere quindi adattata facilmente poiché è modulare. Un altro esempio di questo importante aspetto è rappresentato dal pushbutton "--- Power" presente nella sezione *Compute and Store signal Power* del *Frequency Domain* (area blu), che permette di aggiungere funzionalità di calcolo della potenza in altre bande di interesse, oltre ad *alpha* e *beta*.

Il pulsante rosso "EXIT" chiude l'intera interfaccia al clic del mouse, stampando a video la scritta "*Goodbye*" sul Command Window.

Nel seguito verranno descritte le principali funzionalità implementate e il risultato a video quando l'utente sceglie di eseguire tali funzionalità.

4.2 SCELTA SISTEMA DI ACQUISIZIONE E CARICAMENTO DATI

La primissima cosa da fare è, come già detto, selezionare il sistema di acquisizione utilizzato: Neurowave oppure OpenBCI.

A questo punto, in base alla scelta effettuata, l'utente può trovarsi di fronte a due diverse finestre di dialogo. Anche queste nuove finestre sono state realizzate in ambiente GUIDE, e sono quindi due piccole e semplici GUI a sé stanti.

Se viene selezionato **Neurowave**, compare una finestra di dialogo chiamata *sceltaesperimentoNW* (vedi Figura 4.2) che permette all'utente di selezionare quale esperimento analizzare fra quelli realizzati tramite Neurowave nel laboratorio LIB (completo, mentale, lettura o movimento dito).

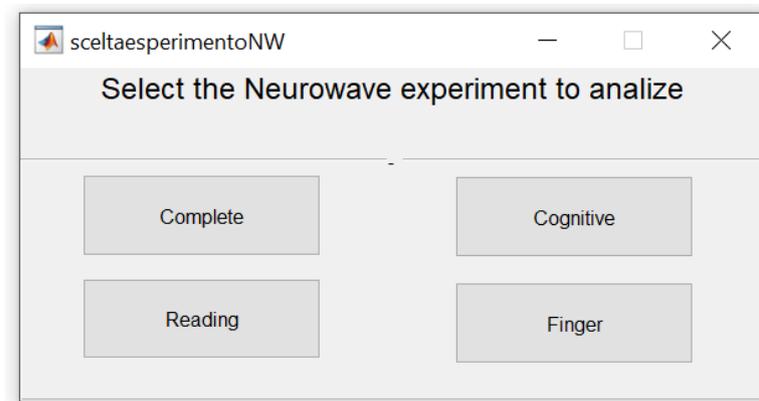


Figura 4.2. Finestra di dialogo per la scelta dell'esperimento Neurowave; (tratta da Matlab attraverso il tasto Stamp)

La scelta di uno di questi quattro pulsanti porta alla comparsa di una finestra esplora-risorse (vedi Figura 4.3).

Questa finestra si apre sul direttorio che contiene i file delle acquisizioni di tutti i soggetti relative a quello specifico esperimento; da questa finestra che l'utente può caricare i dati della specifica prova, ad esempio *nomecognome_reading.mat*.

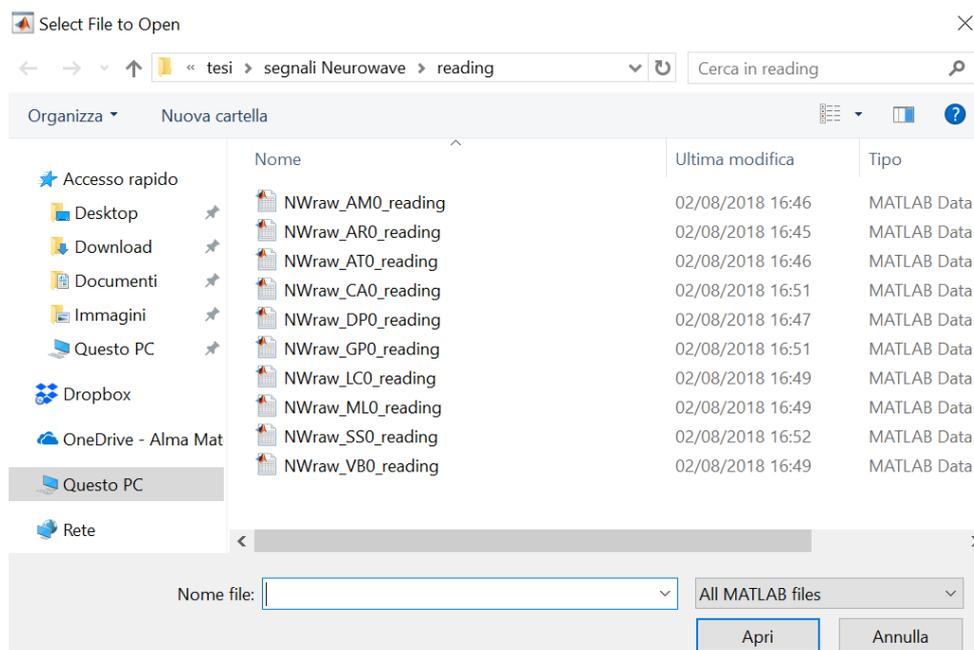


Figura 4.3. Finestra esplora-risorse che compare alla pressione del pulsante 'Reading' in "sceltaesperimentoNW"; (tratta da Matlab attraverso il tasto Stamp)

I dati caricati vengono salvati su Workspace nella variabile di nome *NWchosensignal*. Essa è una variabile $N \times 14$, dove N è il numero di campioni acquisiti e 14 è il numero di canali relativi agli elettrodi (13 per EEG ed uno per ECG).

Se, invece, viene selezionato **OpenBCI**, allora compare una finestra di dialogo chiamata *sceltaesperimentoOBCI* (vedi Figura 4.4) che permette all'utente di scegliere fra esperimento cabina B1 o cabina B2.

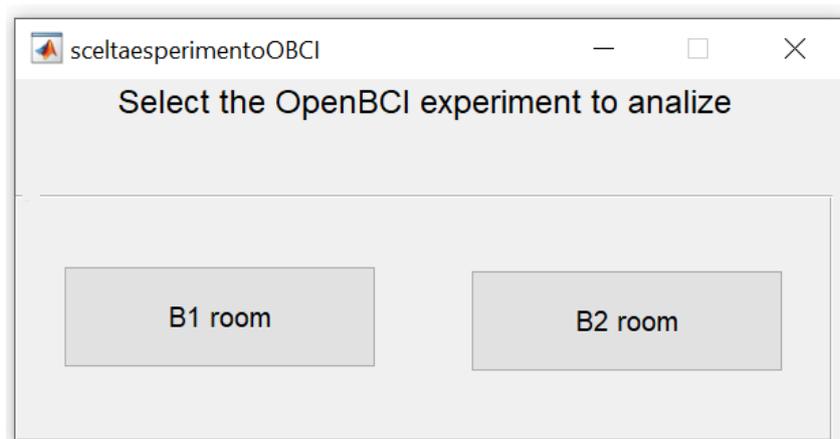


Figura 4.4. Finestra di dialogo per la scelta dell'esperimento OpenBCI; (tratta da Matlab attraverso il tasto Stamp)

La scelta di uno di questi due pulsanti porta anche in questo caso alla comparsa di una finestra esplora-risorse. Questa finestra si apre sul direttorio che contiene i file delle acquisizioni di tutti i soggetti relative a quello specifico esperimento: l'utente può caricare i dati della specifica prova, ad esempio *nomecognome_roomB1.mat*.

I dati caricati vengono salvati su Workspace nella variabile di nome *OBCIchosensignal*. Essa è una variabile $N \times 13$, dove N è il numero di campioni acquisiti e 13 è il numero di canali relativi agli elettrodi (12 per EEG ed uno per ECG). Si ricorda infatti che le acquisizioni eseguite con il sistema OpenBCI veniva acquisito un elettrodo in meno (Cz) rispetto alla acquisizioni eseguite con il sistema Neurowave.

4.3 INSERIMENTO DELLA FREQUENZA DI CAMPIONAMENTO

Una volta caricati i dati di interesse, all'utente spetta indicare la frequenza di campionamento. Ciò è reso possibile dall'edit-box dentro il quale si digita, tramite tastiera, la *sample frequency* (F_s , in Hz). In fase di acquisizione, Neurowave campionava a $F_s=128$ Hz, mentre OpenBCI a $F_s=125$ Hz. L'utente dovrà quindi indicare, a seconda dei casi, una di queste due frequenze. Trattandosi di un Edit Box e non di un Pop Up Menu o List Box (con una lista predefinita di frequenze), all'utente viene lasciata piena libertà nell'indicare la frequenza di campionamento. Infatti, in futuro la GUI potrà essere usata anche per gestire dati acquisiti a frequenza diverse dalle due frequenze indicate.

4.4 VISUALIZZAZIONE EEG NEL DOMINIO DEL TEMPO

L'indicazione della frequenza di campionamento da parte dell'utente permette di definire la scala dei tempi come precedentemente spiegato. Quindi, l'utente accede al *Data selection Panel* e decide se visualizzare un singolo canale o tutti i canali insieme.

Sebbene in molti casi la visualizzazione di tutti i canali sia più adatta per avere un quadro di insieme, può anche risultare utile la visualizzazione di un singolo segnale EEG relativo ad uno specifico canale.

Ciò è reso possibile dal pop-up menu chiamato *Single-channel*: esso è un menu a tendina che contiene, uno di seguito all'altro, i nomi di tutti gli elettrodi utilizzati. La peculiarità è che questo stesso *object* è utilizzato sia per Neurowave che per OpenBCI seppure numero e nome degli elettrodi siano diversi nei due casi: infatti il pop-up menu si popola con il nome dei soli elettrodi relativi all'uno o all'altro sistema di acquisizione, mediante un caricamento di stringa (*load*) da un file Excel chiamato *elenco_elettrodi_utilizzati.xlsx*. Un esempio di visualizzazione singola è mostrato in Figura 4.5, nello specifico si è selezionato l'elettrodo PO7 durante un esperimento (di tipo 'Complete') di Neurowave.

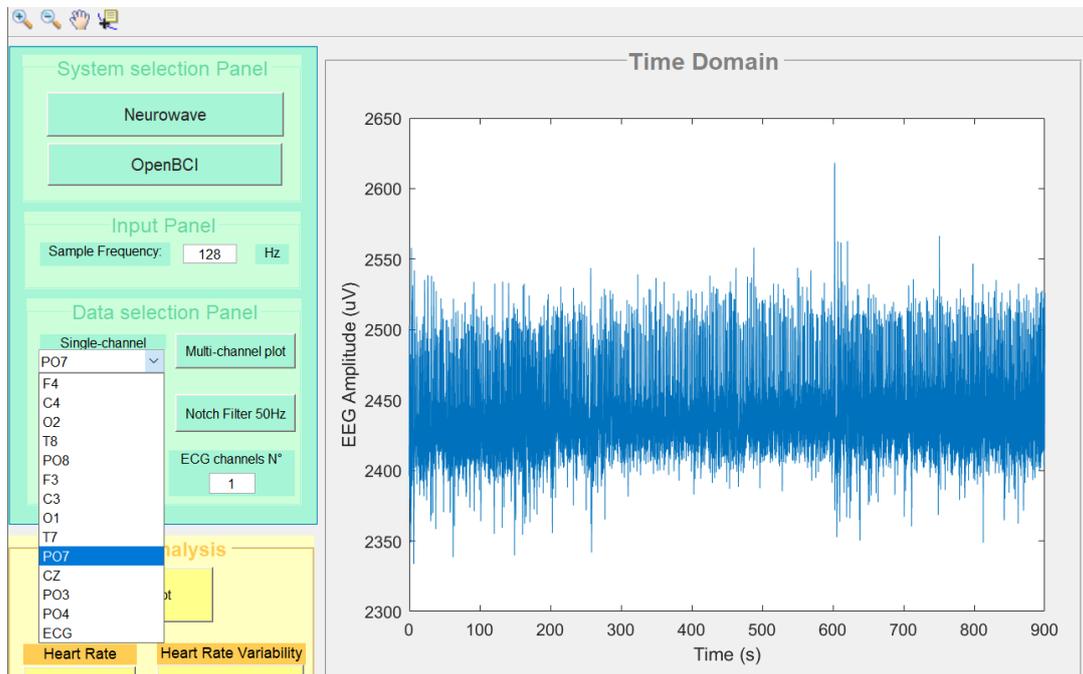


Figura 4.5. Visualizzazione del segnale EEG registrato dal singolo elettrodo PO7 durante un esperimento di tipo 'Complete' in cui era applicato il sistema di acquisizione Neurowave; (tratta da Matlab attraverso il tasto Stamp)

La funzionalità di visualizzazione più significativa del *Data selection Panel* non è la visualizzazione singola, bensì, come già detto, la visualizzazione simultanea di tutti i segnali EEG acquisiti mediante gli elettrodi sulla cuffia. Ciò è possibile grazie al pushbutton *Multi-channel plot*.

Infatti, facendo clic su tale pulsante, vengono visualizzati uno sopra l'altro sull'asse delle ordinate i tracciati EEG di tutti gli elettrodi, in funzione dello stesso asse temporale. Per ottenere questo, è stato usato in Matlab una sola funzione *plot* con cui viene creato un unico grafico all'interno del quale tutti i segnali vengono

visualizzati traslati uno rispetto all'altro lungo l'asse y di un offset fissato. Viene inoltre indicato il nome di ciascun elettrodo in corrispondenza del relativo tracciato, in modo da permettere una facile ed immediata identificazione di questi. Un esempio di multi-channel plot è mostrato in Figura 4.6. Notare il problema della separazione non uniforme dei tracciati, con una tendenza alla sovrapposizione di alcuni di essi nonché delle etichette col nome degli elettrodi (si vedano ad es. i segnali C3 e F3 nel caso illustrato in Figura 4.6). Questa è una conseguenza del fatto che pur imponendo attraverso il codice Matlab una traslazione uniforme tra i segnali, essi presentano un valore medio (non nullo) che può essere diverso da segnale a segnale, così che i segnali appaiono non equispaziati. Il metodo per risolvere questo problema è spiegato nella pagina seguente.

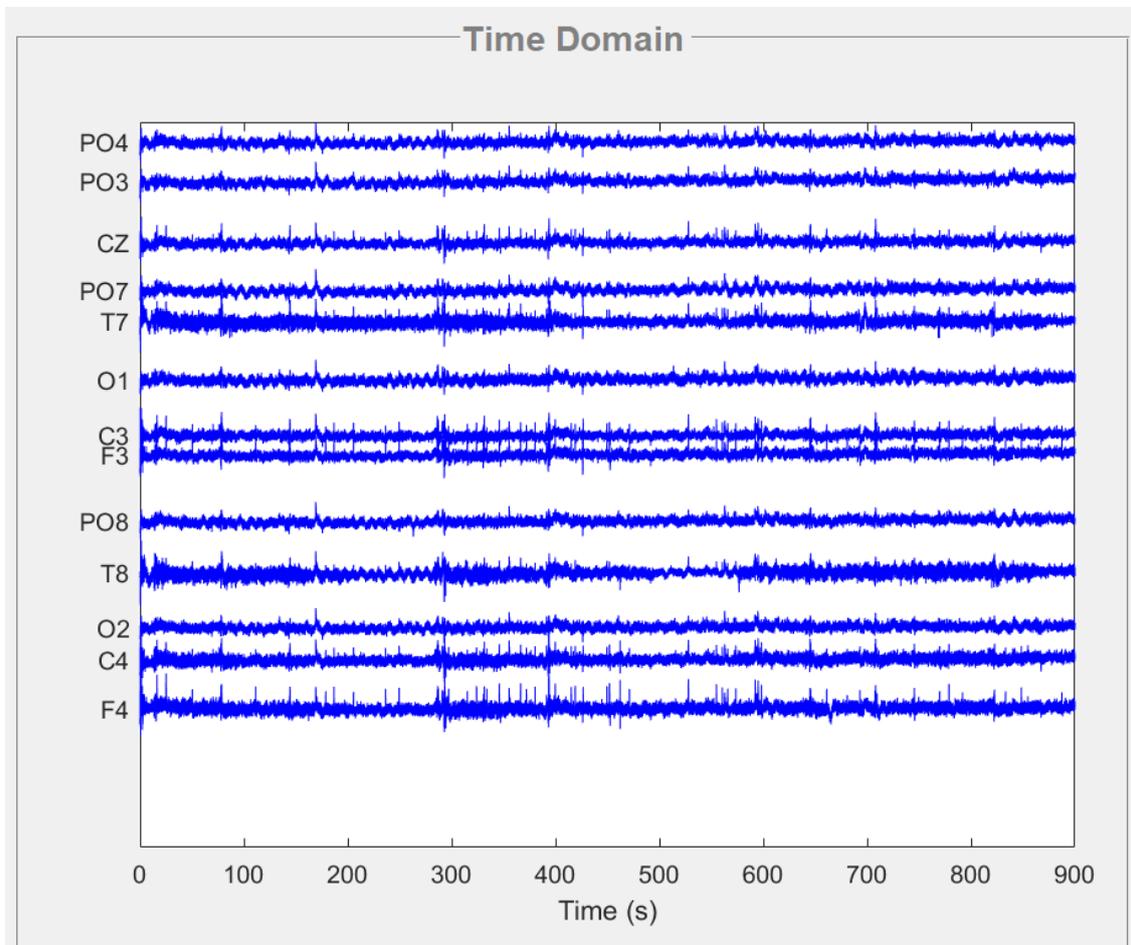


Figura 4.6. Multi-channel Plot, la visualizzazione simultanea di tutti i segnali EEG; (tratta da Matlab attraverso il tasto Stamp)

Al fine di garantire una più appropriata visualizzazione, all'interno del *Data selection Panel* è stata implementata la possibilità di eliminare il valore medio dei segnali, grazie al pushbutton *De-mean*. In questo modo, venuta meno “la continua” di ciascun segnale, tutti i tracciati risultano distanziati fra loro dello stesso valore di offset impostato nel codice Matlab. In Figura 4.7 viene rappresentato il multi-channel plot degli stessi segnali di Figura 4.6 ma dopo aver eliminato il loro valor medio: notare infatti che ora la visualizzazione risulta più chiara ed ordinata, e le etichette dei nomi d'elettrodo non si sovrappongono più.

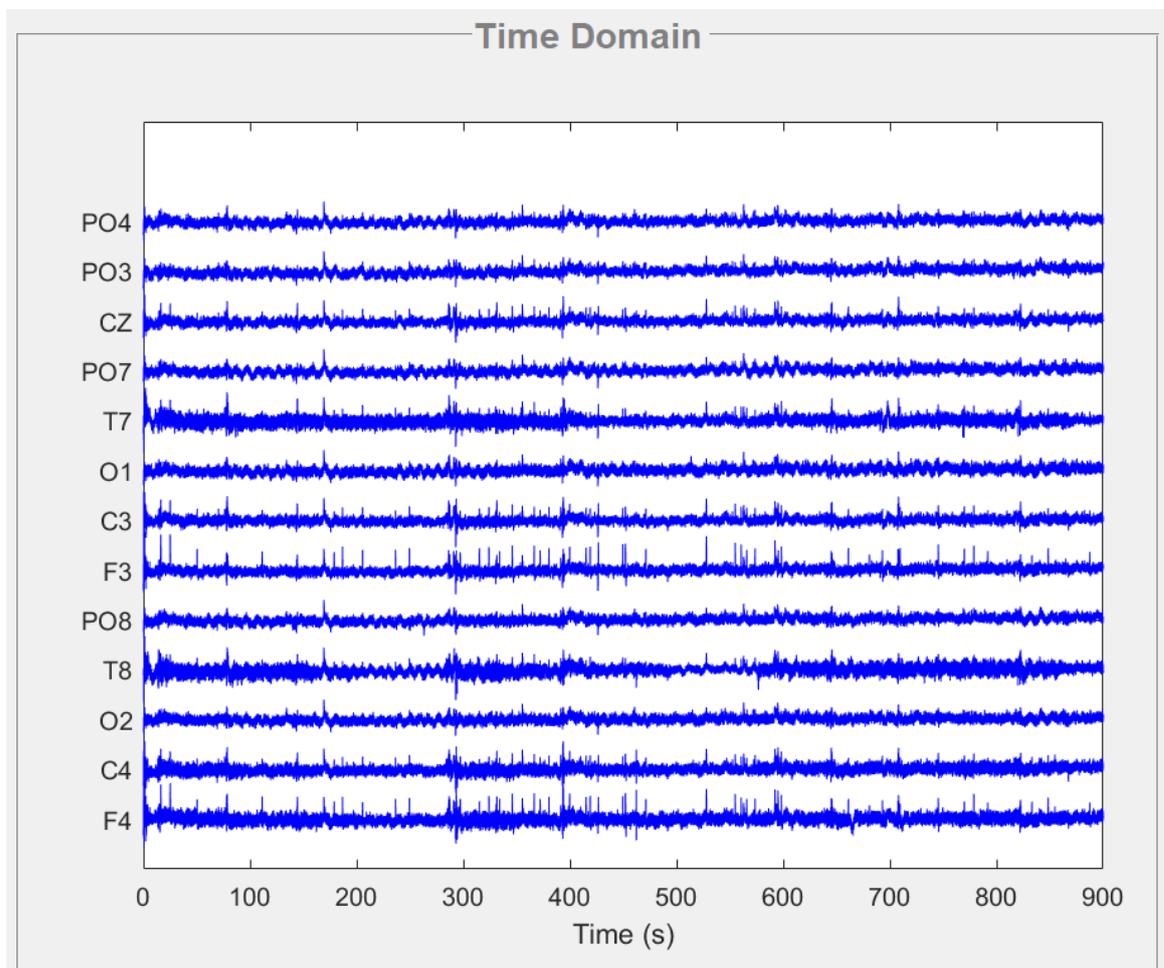


Figura 4.7. Multi-channel Plot De-mean, la visualizzazione simultanea di tutti i segnali EEG al netto del proprio valor medio e con l'aggiunta di un offset progressivo; (tratta da Matlab attraverso il tasto Stamp)

Il numero di canali EEG e canali ECG utilizzati in ciascun esperimento viene aggiornato e mostrato in automatico all'interno dei due edit-box dedicati presenti nella *Data selection Panel*. Così, per Neurowave si visualizzano i numeri '13' (EEG) ed '1' (ECG), mentre per OpenBCI i numeri '12' (EEG) ed '1' (ECG).

4.5 OPERAZIONI DI FILTRAGGIO IN FREQUENZA

Un primo tipo di filtraggio in frequenza è permesso all'interno della *Data selection Panel*. Esso presenta il pushbutton *Notch Filter 50Hz*, che implementa la possibilità di filtrare il segnale d'origine per eliminare la componente di disturbo di rete, tipicamente a 50 Hz (60 Hz in tutta l'America Settentrionale), tramite filtro eliminabanda appunto (vedi Figura 4.8)

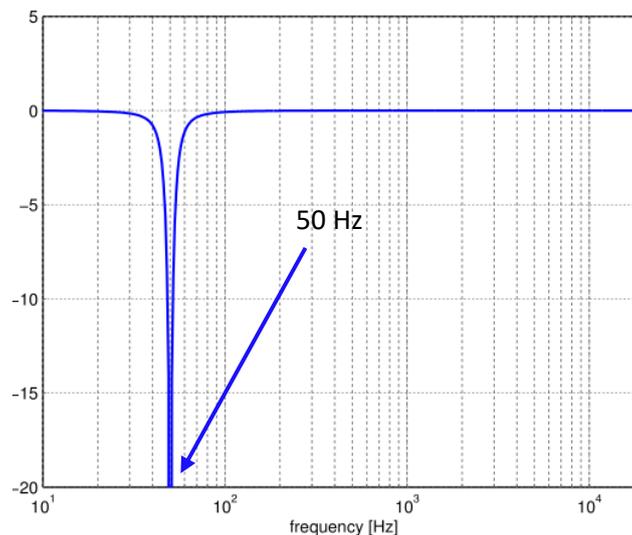


Figura 4.8. Risposta in frequenza di un filtro Notch a 50 Hz; (tratta da [33])

L'utente dispone di altri due tipi di filtraggio, questa volta nell'area blu *Frequency Domain*. Si tratta di filtro *passa-basso* (LPF) e filtro *passa-alto* (HPF).

Questi richiedono la scelta della frequenza di taglio da parte dell'utente attraverso un popup menu che elenca i valori possibili. Una volta scelta la frequenza di taglio del filtro, la selezione del corrispondente pushbutton (LPF o HPF) implementa il codice del filtraggio vero e proprio.

La scelta della frequenza di taglio per il filtraggio passa basso LPF può ricadere sui valori 10Hz, 20Hz, 30Hz, 40Hz e 50Hz; mentre per il filtraggio passa alto HPF sui valori 0.16Hz, 0.30Hz, 0.50Hz, 1.00Hz e 5Hz.

L'azione di LPF ed HPF può essere combinata, al fine di realizzare un filtraggio passa-banda. Ad esempio, selezionando la frequenza di taglio a 0.5 Hz per il filtro passa-alto (HPF) e la frequenza di taglio 20 Hz per il filtro passa-basso (LPF) il segnale EEG viene analizzato limitatamente all'intervallo di frequenze 5-20Hz.

4.6 STIMA DELLA DENSITÀ SPETTRALE DI POTENZA DEL SEGNALE

All'interno della sezione *Frequency Domain*, oltre ai filtri LP e HP appena descritti, sono state implementate le funzionalità più importanti dell'intera GUI per quanto riguarda l'analisi dei segnali EEG registrati. Esse concernono la stima della Densità Spettrale di Potenza del segnale (PSD) ed il calcolo della Potenza del segnale in bande di frequenza specifiche, ad esempio banda *alpha* e banda *beta*.

Per quanto riguarda la PSD del segnale, la GUI offre la possibilità all'utente di valutarla per l'esperimento nella sua interezza oppure separatamente nelle varie fasi dell'esperimento.

La prima cosa da fare per computare la PSD del segnale è impostare la lunghezza (in campioni) della finestra per la PSD. Ciò è possibile grazie all'edit-box *Window*, in cui l'utente inserisce la durata (in secondi) della finestra temporale: l'algoritmo divide il segnale in ingresso in segmenti di lunghezza pari al valore inserito. Successivamente occorre impostare la risoluzione in frequenza per la PSD, il che corrisponde a specificare il numero di punti della Trasformata Discreta di Fourier (DFT) da utilizzare per la stima. Ciò è possibile grazie all'edit-box *Resolution*. Valori tipici per *Window* e *Resolution* sono rispettivamente 10 secondi e 0.1 Hz.

A questo punto è possibile applicare l'algoritmo di stima della PSD del segnale in questione tramite clic sull'apposito pushbutton *Apply PSD*. Il risultato di questa operazione è la generazione di una nuova figura che contiene, grazie alla funzionalità *subplot* di MATLAB, tanti grafici quanti sono gli elettrodi EEG in questione (quindi 12 per OpenBCI e 13 per Neurowave).

Ciascuno di questi grafici è relativo ad un diverso elettrodo e rappresenta la densità spettrale di potenza del segnale acquisito al corrispondente elettrodo. Sull'asse delle ascisse vi sono le frequenze (in Hz) e sull'asse delle ordinate vi è la PSD (espressa in microvolt al quadrato diviso Hz, essendo i segnali acquisiti in microvolt).

Un esempio di tali grafici è mostrato in Figura 4.9, e nello specifico è rappresentato un esperimento di tipo *Reading* di Neurowave.

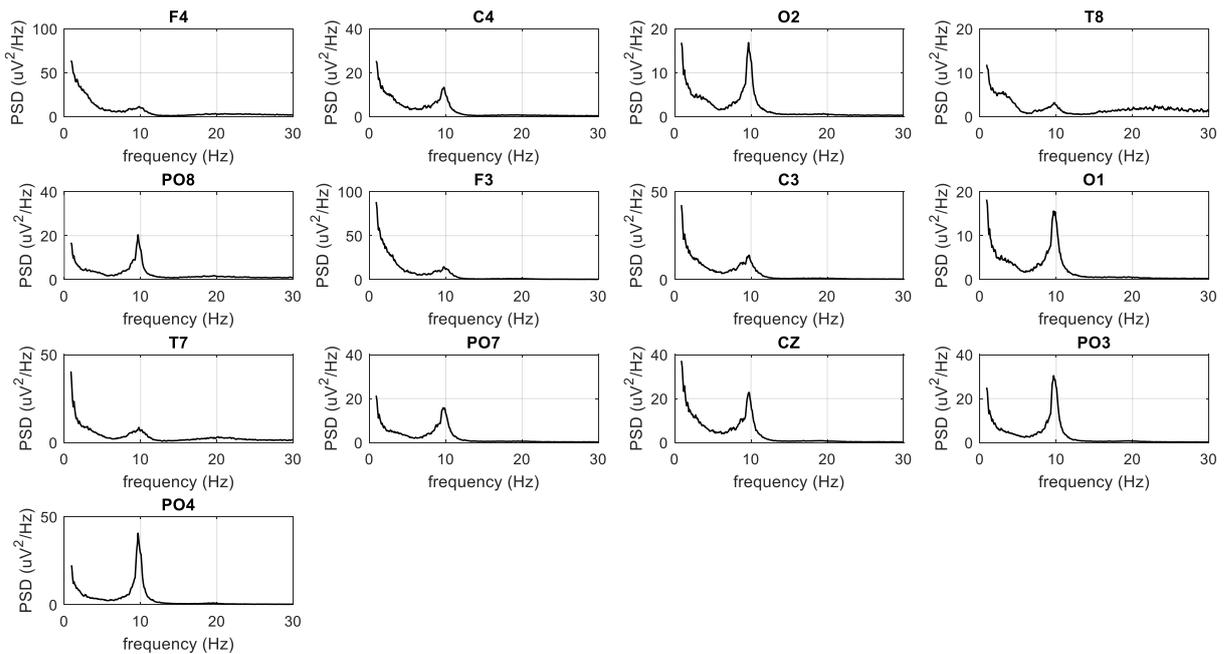


Figura 4.9. Densità spettrale di potenza dei segnali EEG registrati dagli elettrodi durante l'esperimento *Reading* di Neurowave; (tratta da Matlab attraverso la funzione *salva figura*)

La seconda possibilità messa a disposizione dell'utente è quella di stimare la PSD del segnale registrato da ciascun elettrodo separatamente nelle varie fasi dell'esperimento. Pertanto, per gli esperimenti Neurowave vi saranno tre fasi (pre-task, task, post-task) ciascuna da 5 minuti (300 s); mentre per gli esperimenti OpenBCI vi saranno quattro fasi (relax VR off, relax VR on, interazione, final relax VR on) ciascuna da 300 secondi.

Per impostare il numero di fasi e la durata di ciascuna di esse sono stati implementati due ulteriori edit-box, mostrati a titolo d'esempio in Fig 4.10. In Figura 4.11 è mostrato un esempio dei grafici che vengono generati con questa modalità; nella fattispecie si fa riferimento allo stesso esperimento di lettura Neurowave mostrato in Figura 4.9. Come si può osservare dalla Figura 4.11, anche

in questo caso viene creata una figura con un grafico per ciascuna elettrodo. Ogni grafico, mostra la densità spettrale di potenza del segnale all'elettrodo corrispondente usando colori distinti per ciascuna fase.

Power Density Spectrum step-by-step experiment	
Steps N°	Step duration (s)
3	300
Window (s)	Resolution (Hz)
10	0.1
Apply PSD	

Figura 4.10. Pannello di inserimento dei valori per PDS differenziata per le varie fasi dell'esperimento; (tratta da Matlab attraverso il tasto Stamp)

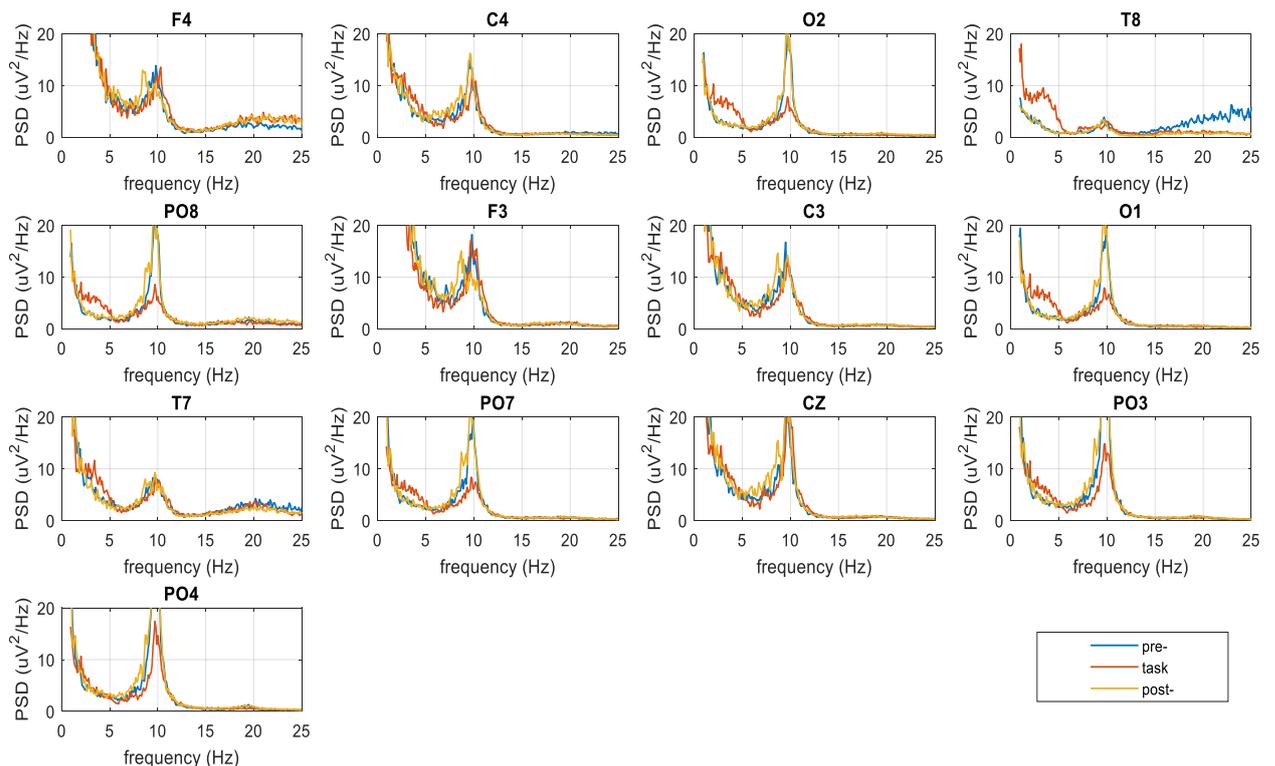


Figura 4.11. Densità spettrale di potenza dei segnali EEG registrati dagli elettrodi durante esperimento lettura Neurowave differenziata per le tre fasi (mostrate in legenda). In ogni grafico, sono visualizzate tre linee con tre colori distinti che rappresentano la PSD del segnale all'elettrodo corrispondente in ciascuna delle tre fasi dell'esperimento (Tratta da Matlab attraverso la funzione salva figura).

4.7 CALCOLO DELLA POTENZA DEL SEGNALE IN SPECIFICHE BANDE

Dopo aver valutato e visualizzato la PSD dei segnali EEG in oggetto, l'utente ha la possibilità di proseguire l'analisi in frequenza calcolando la potenza di tali segnali in bande di frequenza specifiche. A tal fine, come mostrato in Figura 4.12, sono stati progettati due pushbutton, più un terzo per scopi futuri, che implementano il calcolo della suddetta potenza nelle due bande di frequenza significative per questo progetto, *alpha* e *beta*, salvano i valori di potenza così calcolati in una apposita cartella (in modo da poter essere utilizzati per ulteriori analisi anche al di fuori della GUI), e generano una serie di grafici, descritti nel seguito.

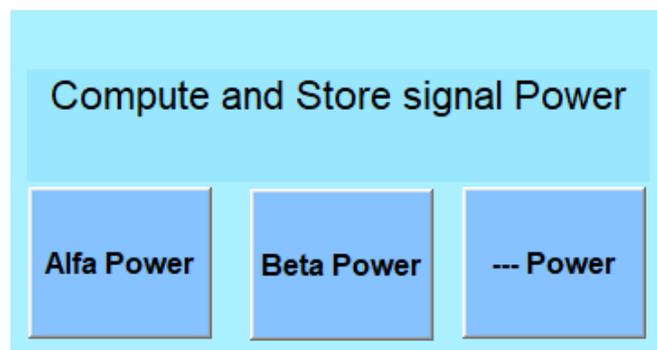


Figura 4.12. Pannello per il calcolo e salvataggio della potenza del segnale in banda *alpha* e *beta*. Notare il terzo pushbutton di funzione non ancora definita, dedicato a scopi futuri; (tratta da Matlab attraverso il tasto Stamp)

Il risultato del codice di controllo del pushbutton “Alpha Power” è quello di generare un vettore, chiamato *alfa_power_vector*, contenente i valori di potenza *alpha* del segnale per ciascun elettrodo e per ciascuna fase, per un totale di 39 elementi nel caso di Neurowave (13 elettrodi EEG per tre fasi ciascuno) o di 36 elementi nel caso di OpenBCI (12 elettrodi EEG per tre fasi ciascuno, trascurando la fase di interazione con la VR).

Come mostrato a titolo d'esempio in Figura 4.13, tali valori di potenza vengono rappresentati, tramite *subplot*, in modo che gli asterischi blu, in ciascuno dei grafici d'elettrodo, rappresentino nell'ordine le tre fasi analizzate (da sinistra verso destra): pre-task, task, post-task.

È caratteristico notare come la potenza *alpha* subisca un abbassamento nella fase centrale in ciascun elettrodo, ovvero la fase di task, ed un incremento nella fase finale, ovvero di rilassamento a fine esercizio.

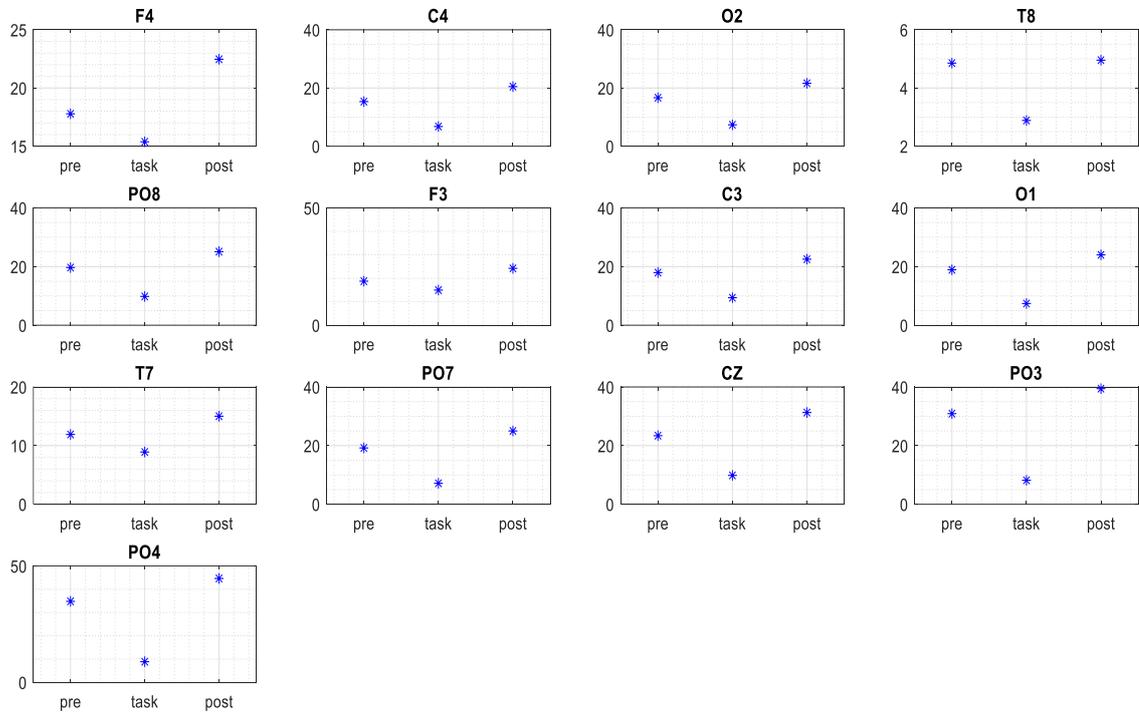


Figura 4.13. Rappresentazione grafica della potenza *alpha* registrata da ciascun elettrodo nelle tre fasi dell'esperimento "lettura" di Neurowave. (Tratta da Matlab attraverso la funzione *salva figura*).

Una ulteriore analisi che è possibile condurre riguardo al tema della potenza dei segnali, è calcolare un valore di potenza medio per ciascuna delle cinque zone dello scalpo su cui si è concentrata l'acquisizione dei segnali, e per ciascuna fase. In altre parole, si esegue una media fra i valori di potenza ottenuti in ciascuna fase per gli elettrodi *frontali* (F3, F4), *centrali* (C3, C4 e, solo nel caso Neurowave, Cz), *temporali* (T7, T8), *parieto-occipitali* (PO3, PO4, PO7, PO8) e *occipitali* (O1, O2).

In questo modo si ottengono tre valori di potenza *alpha* per ciascuna delle cinque aree d'interesse, rappresentativi delle tre fasi dell'esperimento.

I risultati dell'analisi appena descritta sono rappresentati in un grafico che raffigura le cinque curve di variabilità della potenza *alpha* in dipendenza delle fasi dell'esperimento. Un esempio di tale grafico è mostrato in Figura 4.14.

Una simile rappresentazione fornisce un utile strumento sia per valutare la variabilità della potenza del segnale registrato in una zona dello scalpo al variare della fase dell'esperimento, sia per attuare un confronto fra zone diverse; in particolare, la figura sottostante mostra che le zone più sollecitate dal punto di vista del ritmo *alpha* sono, come ci si aspettava, la occipitale (verde) e la parieto-occipitale (viola), mentre la regione meno coinvolta dal ritmo *alpha* è quella temporale.

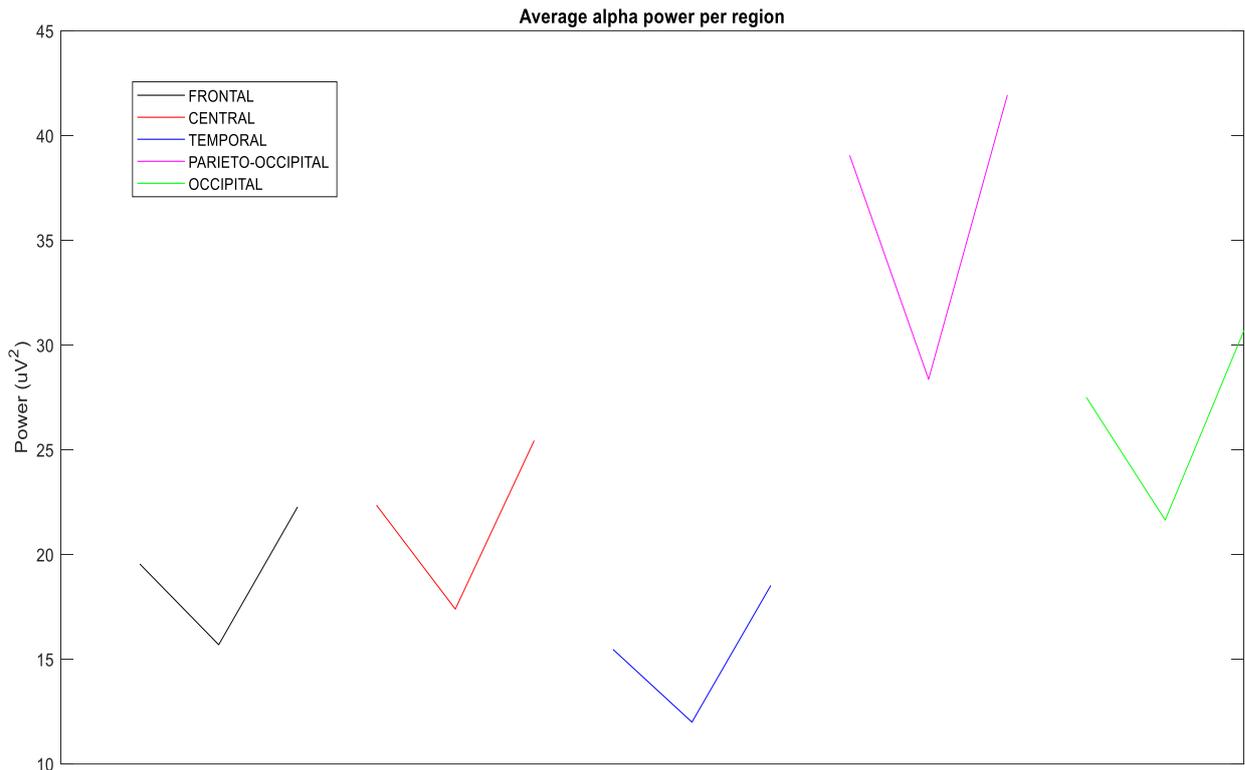


Figura 4.14. Curve di variabilità, nelle tre fasi dell'esperimento, della potenza *alpha* media per regione dello scalpo; nello specifico: esperimento lettura Neurowave. (Tratta da Matlab attraverso la funzione `salva figura`).

Per completezza, si presentano i grafici ed i risultati ottenuti alla pressione del pushbutton “Alpha Power” nella GUI avendo selezionato un segnale di OpenBCI invece di Neurowave.

In particolare, la Figura 4.15 è la rappresentazione grafica della potenza *alpha* del segnale registrato da ciascun elettrodo nelle tre fasi dell'esperimento: relax VR off (r1), relax VR on (r1VR), final relax VR on (r2VR); mentre la Figura 4.16 rappresenta le curve di variabilità, nelle tre fasi dell'esperimento, della potenza *alpha* media per regione dello scalpo.

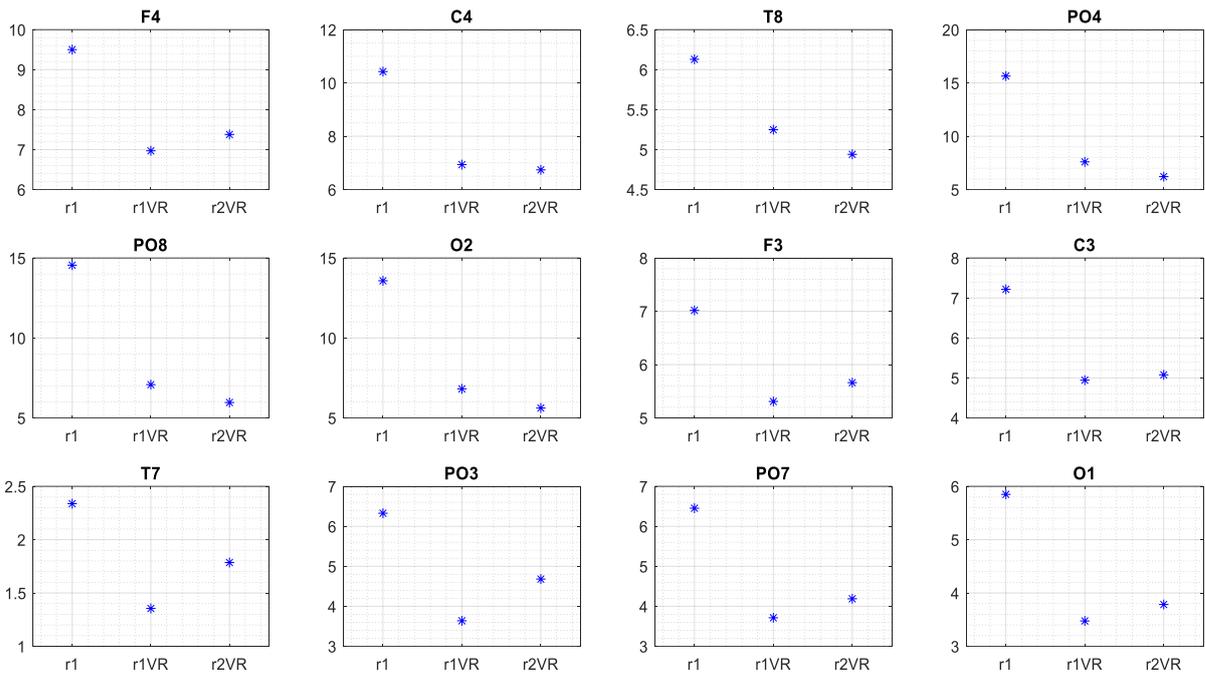


Figura 4.15. Rappresentazione grafica della potenza alpha registrata da ciascun elettrodo in tre fasi dell'esperimento "cabina B2" di OpenBCI. (Tratta da Matlab attraverso la funzione salva figura).

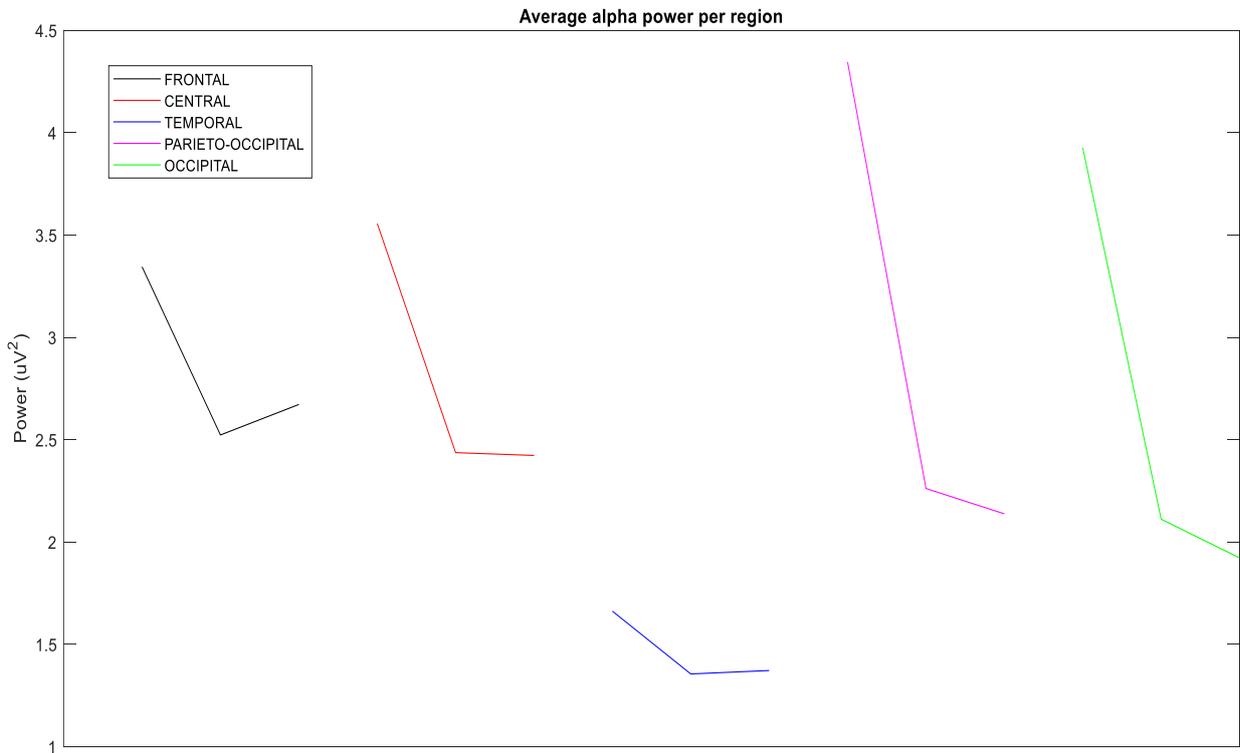


Figura 4.16. Curve di variabilità, in tre fasi dell'esperimento, della potenza alpha media per regione dello scalpo; nello specifico: esperimento "cabina B2" OpenBCI. (Tratta da Matlab attraverso la funzione salva figura).

4.8 VISUALIZZAZIONE DEL SEGNALE ECG E ANALISI DELL'HRV

L'area della GUI contraddistinta dallo sfondo di colore giallo è dedicata all'analisi elettrocardiografica del segnale acquisito durante gli esperimenti ponendo due elettrodi sulla cute del soggetto volontario poco al di sotto delle clavicole. Il pushbutton "Plot" carica il segnale ECG relativo alla prova selezionata, crea un vettore temporale con lo stesso meccanismo illustrato in precedenza, e genera un grafico tempo-ampiezza in una figura separata dai segnali EEG, in modo da permettere una più appropriata visualizzazione (vedi Figura 4.17a). Grazie agli strumenti di zoom e traslazione messi a disposizione direttamente da Matlab, l'utente può selezionare una parte del segnale ECG limitatamente ad un certo intervallo temporale ed eseguire un ingrandimento sulla stessa; un esempio di questa operazione è mostrato in Figura 4.17b.

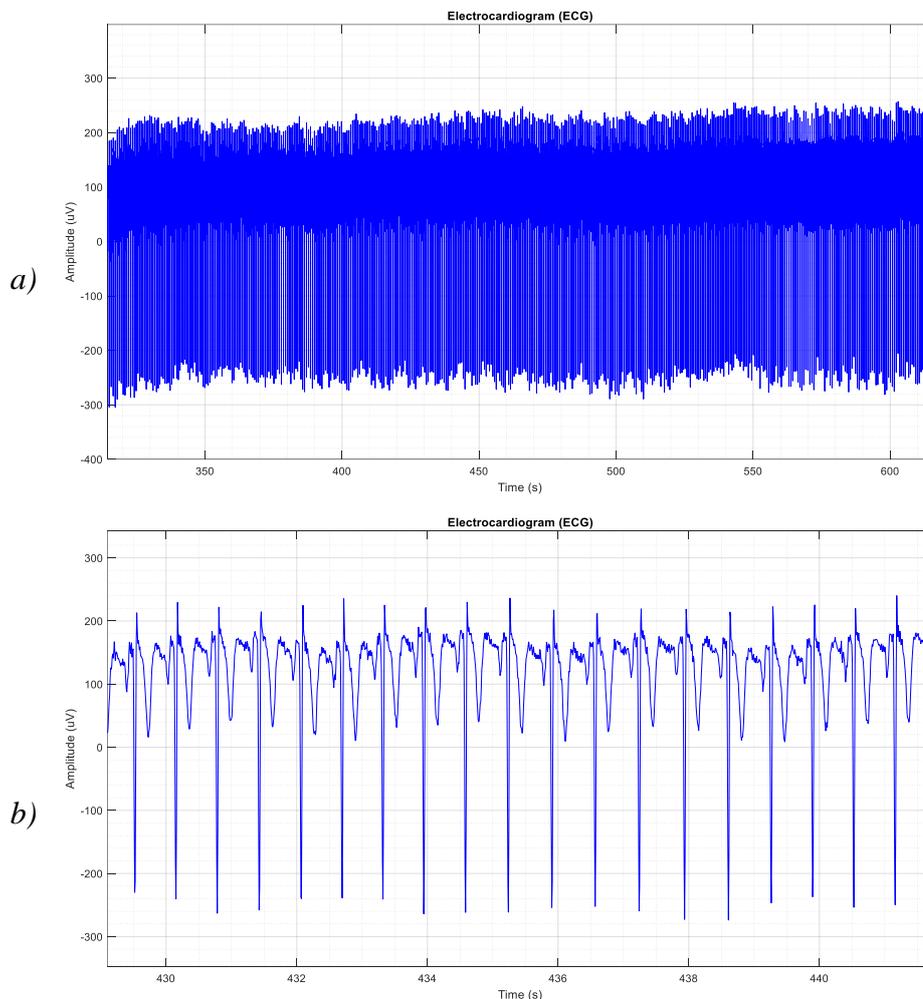


Figura 4.17. Andamento del segnale ECG nel tempo (a) e zoom sul segnale in un intervallo di circa 15 secondi (b). (Tratta da Matlab attraverso la funzione *salva figura*).

Dopo aver eseguito una visualizzazione del segnale ECG nel tempo, l'utente può ora passare alle successive fasi dell'analisi elettrocardiografica, che sono implementate, nell'ordine, dai pushbutton "Tachogram", "HR" ed "HRV".

Il pushbutton "Tachogram" realizza il tacogramma del segnale ECG suddetto. Si ricorda che con il termine tacogramma ci si riferisce al grafico di una sequenza di intervalli temporali fra le onde R di battiti successivi, pertanto esso può essere dato come tempo-su-tempo oppure tempo-su-numero di battiti. Si osservi che in questo lavoro non è stato implementato il codice per estrarre dal segnale ECG il tacogramma; questo richiederebbe infatti un algoritmo di riconoscimento dei picchi R dal segnale ECG.

Tali algoritmi sono in taluni casi implementati direttamente dai sistemi di acquisizione del segnale ECG; è il caso, per esempio, del sistema Neurowave il cui software consente di estrarre dal segnale ECG gli intervalli RR. In altri casi, sono disponibili e scaricabili gratuitamente da Internet, software specificamente dedicati all'estrazione degli intervalli RR dai segnali ECG.

I segnali ECG acquisiti attraverso OpenBCI (sistema che di per sé non implementa la funzionalità di estrazione del tacogramma) sono stati preventivamente elaborati – in un lavoro eseguito da un altro tesista – attraverso un software di estrazione del tacogramma.

Pertanto, nella presente GUI si assume che il tacogramma si già stato fornito come dato in ingresso. In Figura 4.18 è mostrato un esempio di tacogramma (sia in funzione del tempo che del numero di battiti), nello specifico si tratta di un esperimento di tipo *Complete* di Neurowave.

Notare che si possono distinguere le tre fasi da circa 300 secondi ciascuna: nei primi 300s del tracciato (relax pre-task) gli intervalli temporali fra picchi R-R si mantengono ad un valore medio di circa 0.9s (frequenza cardiaca di circa 1.11 bps), mentre nella parte di tracciato da 300s a 600s, e quindi in corrispondenza della fase d'esercizio (task), si ha un calo drastico di tale valor medio (e quindi un aumento della frequenza cardiaca), a testimonianza del fatto che eseguire operazioni matematiche rimanendo concentrati per rispondere correttamente e nel minor tempo

possibile, genera uno stato di agitazione nel soggetto, soprattutto nella fase iniziale del task.

Verso la parte finale del task e nella fase di relax post-task il valore medio degli intervalli RR tende a riportarsi ai valori di relax pre-task, pur restando leggermente al di sotto.

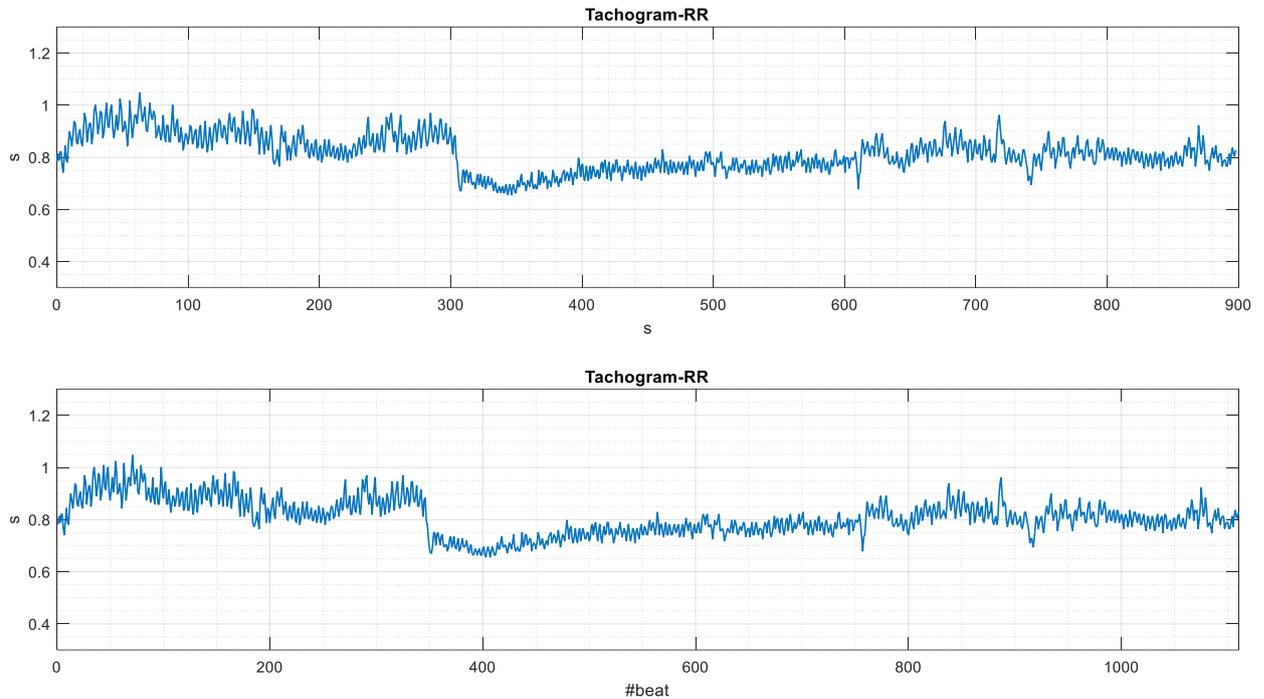


Figura 4.18. Tacogramma in funzione del tempo (sopra) e in funzione del numero di battiti (sotto) realizzato per il segnale ECG rilevato durante un esperimento di tipo “Complete” in Neurowave. (Tratta da Matlab attraverso la funzione salva figura).

Il pushbutton “HR” calcola la frequenza cardiaca a partire dal segnale di Inter-Beat Interval (IBI) utilizzato per il tacogramma e facendone il reciproco. Ad esempio, eseguendo il reciproco di ciascun elemento del vettore che rappresenta un segnale di IBI [0.8 0.65 0.75 0.82] s, si ottiene il vettore [1.25 1.54 1.33 1.22] Hz i cui elementi rappresentano i valori di frequenza cardiaca corrispondenti.

Così facendo viene generato un grafico della frequenza cardiaca in funzione del tempo, mostrato nella parte alta di Figura 4.19.

Inoltre, il codice di controllo del pushbutton HR implementa una seconda funzionalità: il segnale di Heart Rate viene interpolato linearmente con frequenza di 10 Hz e filtrato tramite filtro passa-basso con banda passante fino a 0.5 Hz; alla

fine di tali operazioni il segnale viene nuovamente rappresentato in un grafico HR-tempo, come mostrato nella parte bassa di Figura 4.19.

Parallelamente a quanto visto per i tracciati del tacogramma, i tracciati di HR mostrano repentino cambiamento della frequenza cardiaca al cambiare della fase dell'esperimento, in particolare nel passaggio dalla fase di relax pre-task alla fase di task: a meno di oscillazioni, nei primi 300 secondi della prova la HR permane ad un valore medio di circa 1.1 bps, per poi innalzarsi bruscamente durante la fase di task, ed infine diminuire nuovamente nella fase di relax finale, pur rimanendo a valori più alti rispetto a quelli di relax iniziale.

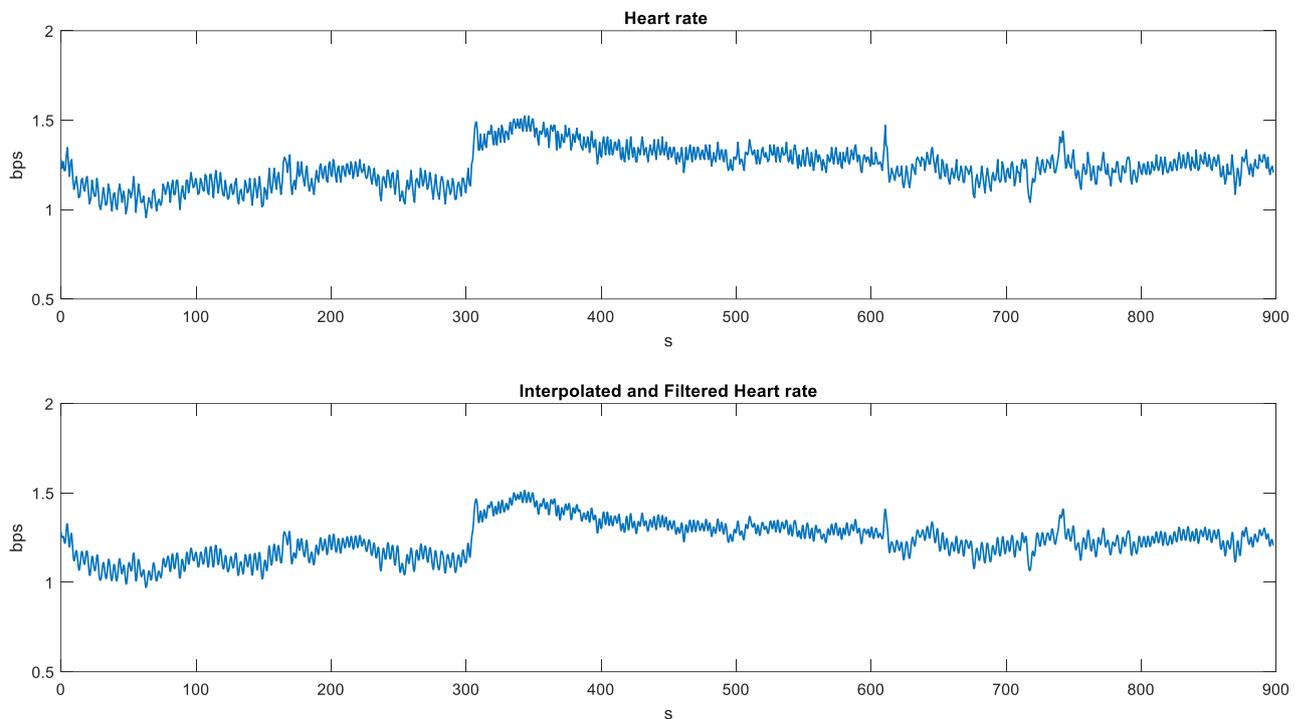


Figura 4.19. Grafico della frequenza cardiaca in funzione del tempo (sopra) e della stessa in seguito alle operazioni di interpolazione lineare e filtraggio passa basso (sotto). (Tratta da Matlab attraverso la funzione salva figura).

Il pushbutton “HRV” implementa le funzionalità più importanti dell’analisi elettrocardiografica dal punto di vista frequenziale. Esso calcola lo spettro di potenza del segnale di HR precedentemente interpolato e filtrato; realizza il grafico della densità spettrale di potenza per le tre fasi dell’esperimento (vedi Figura 4.20); calcola la potenza del segnale nelle bande Low Frequency (LF, 0.04-0.15Hz) ed High Frequency (HF, 0.15-0.4 Hz) come area sottesa alla curva; salva i valori di tali potenze per ciascun soggetto e per ciascuna fase in una cartella apposita, ed infine calcola e stampa su Command Window i valori del rapporto LF/HF nelle tre fasi.

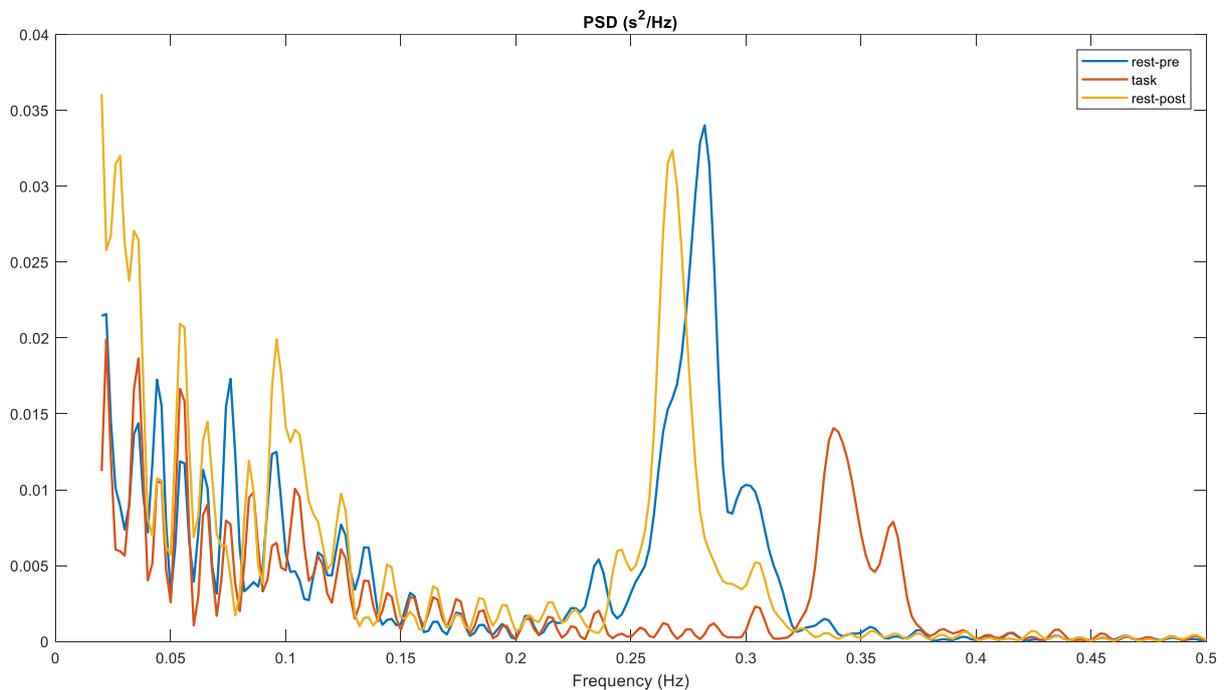


Figura 4.20. Densità spettrale di potenza del segnale di Heart Rate interpolato e filtrato, relativo ad un esperimento di tipo “Complete” di Neurowave. (Tratta da Matlab attraverso la funzione salva figura).

CAPITOLO 5

CODICE MATLAB DI IMPLEMENTAZIONE DELL' INTERFACCIA GRAFICA UTENTE

Nel capitolo precedente sono state descritte la struttura e le funzionalità della GUI dal punto di vista dell'utilizzatore. È stato cioè presentato una sorta di 'manuale d'uso' per utenti non esperti di Matlab (o comunque non interessati al codice Matlab sottostante) i quali vedono il sistema come una sorta di 'scatola nera' in cui sono di interesse solo le interfacce, i valori dei parametri da immettere tramite le interfacce e gli output grafici e numerici restituiti. In questo Capitolo, invece, la 'scatola nera' viene aperta e ne viene mostrato il contenuto: viene, cioè, illustrato il codice Matlab che implementa le principali funzionalità della GUI descritte nel capitolo precedente.

5.1 SELEZIONE DEL SISTEMA DI ACQUISIZIONE

Alla pressione di uno dei due pulsanti (Neurowave od OpenBCI) tramite clic, viene richiamato il codice di controllo contenuto nella relativa funzione di *Callback*. Le due funzioni sono strutturalmente identiche e, dal momento che la loro attivazione è mutuamente esclusiva, al loro interno possono essere assegnate variabili con lo stesso nome.

```
% --- Executes on button press in neurowavebutton.
function neurowavebutton_Callback(hObject, eventdata, handles)
    handles.setnumber=0
    sceltaesperimentoNW
    uiwait
    handles.data=evalin('base','NWchosensignal')
    handles.labels={'F4','C4','O2','T8','P08','F3','C3','O1','T7','P07',
    'CZ','P03','P04'}
    guidata(hObject, handles);

% --- Executes on button press in openbcibutton.
function openbcibutton_Callback(hObject, eventdata, handles)
    handles.setnumber=1
    sceltaesperimentoOBCI
    uiwait
    handles.data=evalin('base','OBCIchosensignal')
```

```
handles.labels={'F4','C4','T8','P04','P08','O2','F3','C3','T7','P03','P07','O1'};  
guidata(hObject, handles);
```

Il primo esempio di questo concetto è dato dalla variabile chiamata *setnumber*, pensata in modo da assumere il valore 0 quando è selezionato Neurowave ed il valore 1 quando è selezionato OpenBCI. Infatti, nel seguito del codice dell'intera GUI, la variabile *setnumber* conterrà solamente il valore 0 o solamente il valore 1 fino a che l'utente non decida di cambiare sistema di acquisizione cliccando sull'altro pulsante.

Dopo aver assegnato il valore alla variabile *setnumber* in base al sistema selezionato, viene richiamata l'esecuzione della finestra di dialogo dedicata alla scelta dell'esperimento (*sceltaesperimentoNW* o *sceltaesperimentoOBCI*), il cui codice è riportato alla fine del presente sotto-capitolo. A questo punto con il comando *uiwait* si comunica alla GUI di interrompere la propria esecuzione ed attendere la scelta dell'utente, ovvero aspettare che egli selezioni uno dei quattro (due) pulsanti dell'esperimento Neurowave (OpenBCI) comparsi nella finestra di dialogo relativa (si vedano le Figure 4.2 e 4.4 del capitolo precedente). Il comando duale di *uiwait* si chiama *uiresume*: contenuto all'interno del codice delle suddette finestre di dialogo, esso permette di riprendere l'esecuzione della GUI precedentemente interrotta.

A questo punto, dopo l'operazione di caricamento dati (che viene eseguita dalla function *sceltaesperimentoNW* o *sceltaesperimentoOBCI*), viene creata la variabile *labels* che contiene le etichette di tutti gli elettrodi utilizzati nei rispettivi sistemi di acquisizione; tale *labels* viene salvata all'interno della struttura *handles*, che contiene i puntatori a tutti gli oggetti della GUI, e permette interoperabilità e condivisione dei dati fra di essi. L'ultima istruzione *guidata(hObject, handles)* salva la struttura *handles* con gli ultimi puntatori aggiunti: è una linea di codice molto importante perché permette un aggiornamento continuo.

Di seguito, a titolo d'esempio, si riporta la parte di codice della funzione *sceltaesperimentoOBCI* che permette il caricamento dati.

In particolare, il codice di controllo dei due pushbutton per la selezione della cabina B1 o B2.

```
% --- Executes on button press in blroom.
function blroom_Callback(hObject, eventdata, handles)
    chosen_experiment=5;
    assignin('base','chosen_experiment',chosen_experiment)
    handles.blroom_data=uigetfile('C:\Users\Gianlu\Documents\tesi\segnali OpenBCI\B1 room')
    assignin('base','stringa_nome_soggetto',handles.blroom_data)
    datastruct=load(handles.blroom_data);
    handles.OBCIchosensignal=datastruct.data
    assignin('base','OBCIchosensignal',handles.OBCIchosensignal)
    guidata(hObject, handles);
    uiresume
    close(gcf);

% --- Executes on button press in b2room.
function b2room_Callback(hObject, eventdata, handles)
    chosen_experiment=6;
    assignin('base','chosen_experiment',chosen_experiment)
    handles.b2room_data=uigetfile('C:\Users\Gianlu\Documents\tesi\segnali OpenBCI\B2 room')
    assignin('base','stringa_nome_soggetto',handles.b2room_data)
    datastruct=load(handles.b2room_data);
    handles.OBCIchosensignal=datastruct.data
    assignin('base','OBCIchosensignal',handles.OBCIchosensignal)
    guidata(hObject, handles);
    uiresume
    close(gcf);
```

Dopo aver creato una variabile *chosen_experiment* identificativa dell'esperimento selezionato, se ne salva il valore su Workspace grazie all'istruzione *assignin*; poi, tramite l'istruzione *uigetfile*, viene aperta la finestra esplora-risorse di selezione del file (analoga a quella mostrata in Figura 4.3). Quando l'utente seleziona il file tramite clic, viene salvata la stringa del percorso a tale file, in questo modo, grazie all'istruzione *load*, si carica tale file e lo si assegna alla variabile *datastruct*. A questo punto, si crea la variabile *OBCIchosensignal* contenente il campo *data* della struttura *datastruct*, e lo si salva su Workspace. Per finire, si salva la struttura degli *handles* grazie all'istruzione *guidata()*, si consente alla GUI precedentemente interrotta di riprendere la propria esecuzione grazie all'istruzione *uiresume*, ed infine si chiude la finestra di dialogo apertasi per la selezione dell'esperimento grazie all'istruzione *close(gcf)*.

5.2 CREAZIONE POPUP MENU DEGLI ELETTRODI E VISUALIZZAZIONE SINGLE-CHANNEL

Dopo la scelta del sistema di acquisizione e del tipo di esperimento, l'utente può passare alla visualizzazione dei segnali nel dominio dei tempi: come spiegato nel Capitolo 4, l'utente ha la possibilità di operare una visualizzazione single-channel oppure multi-channel plot. La porzione di codice sotto riportata è relativa alla visualizzazione del singolo canale.

```
% --- Executes on selection change in pescadaexcel.
function pescadaexcel_Callback(hObject, eventdata, handles)
    if(handles.setnumber==0)
        [~, ~, raw] = xlsread('C:\Users\Gianlu\Documents\tesi\elenco
elettrodi usati.xlsx', 'Foglio1', 'A5:A18');
        stringVectors = string(raw(:,1));
        stringVectors(ismissing(stringVectors)) = '';
        handles.nome_elettrodiNW = stringVectors(:,1);
        nome_elettrodiNW=handles.nome_elettrodiNW;
        set(handles.pescadaexcel, 'String', nome_elettrodiNW)
        data = handles.data;
        str = get(hObject, 'String');
        val = get(hObject, 'Value');
        switch str{val};
            case 'F4' handles.current_data = data(:,1);
            case 'C4' handles.current_data = data(:,2);
            case 'O2' handles.current_data = data(:,3);
            case 'T8' handles.current_data = data(:,4);
            case 'P08' handles.current_data = data(:,5);
            case 'F3' handles.current_data = data(:,6);
            case 'C3' handles.current_data = data(:,7);
            case 'O1' handles.current_data = data(:,8);
            case 'T7' handles.current_data = data(:,9);
            case 'P07' handles.current_data = data(:,10);
            case 'CZ' handles.current_data = data(:,11);
            case 'P03' handles.current_data = data(:,12);
            case 'P04' handles.current_data = data(:,13);
            case 'ECG' handles.current_data = data(:,14);
        end
        guidata(hObject, handles);
        tempi=(1:length(handles.current_data))./handles.fs;
        plot(tempi,handles.current_data)
        xlim([0 900]), xlabel('Time (s)'), ylabel('EEG Amplitude (uV)')
    else
        [...]
```

La prima cosa che il codice fa è capire se è stato premuto il pulsante Neurowave oppure OpenBCI testando il valore della variabile *setnumber* (costrutto “*if...else*”). Il codice contenuto all’interno del costrutto *if-else* è strutturalmente lo stesso per i due sistemi di acquisizione, ma caratterizzato da termini ed elettrodi diversi; pertanto si continua la trattazione con il solo caso Neurowave, a titolo di esempio.

Le prime tre righe del codice sono relative al caricamento delle stringhe coi nomi degli elettrodi utilizzati da un file Excel, e tale vettore stringa è salvato nella variabile *nomi_elettrodiNW*. Con la successiva istruzione di *set* si impostano tali nomi ad essere le stringhe dell’oggetto *pescadaexcel* (che è un object di tipo popup menu), in tal modo il popup menu risulta popolato con le corrette etichette d’elettrodo (*labels*). A questo punto, tramite il costrutto “*switch...case*”, si realizza il codice di controllo relativo a ciascuna delle scelte possibili all’interno del suddetto popup menu, ovvero cosa il codice deve fare alla selezione di una delle voci della tendina. La variabile *data* ($N \times 14$) contiene i dati dei segnali acquisiti durante l’esperimento: N è il numero di campioni, 14 è il numero di canali (ogni colonna corrisponde ad un elettrodo EEG, l’ultima colonna corrisponde alla derivazione ECG). Selezionando una delle voci del popup menu, viene creata ed aggiunta alla struttura *handles* una variabile chiamata *current_data*, ottenuta estraendo dalla variabile *data* la colonna corrispondente allo specifico elettrodo selezionato. Una volta chiuso il costrutto *switch*, il codice aggiorna la struttura degli *handles* e crea il vettore dei tempi dividendo un vettore che va da 1 a N per la frequenza di campionamento. A questo punto è il momento della visualizzazione: tramite l’istruzione *plot* si genera il grafico *tempi-current_data* e si aggiungono i nomi agli assi delle ascisse e delle ordinate.

5.3 ELIMINAZIONE DEL VALOR MEDIO E FILTRAGGIO NOTCH

In questa sezione si analizzano i codici relativi ai pushbutton *De-mean*, per l’eliminazione del valor medio dei segnali, e *Notch filter 50Hz*, per l’eliminazione del disturbo di rete.

```
% --- Executes on button press in demean_pushbutton.  
function demean_pushbutton_Callback(hObject, eventdata, handles)  
    data=handles.data;  
    [m,n]=size(data);
```

```

M=mean(data, 1);
data_demean=data-ones(m, 1)*M;
data=data_demean; % samples x channels
handles.data=data;
assignin('base', 'EEGdemean', data_demean)
guidata(hObject, handles);

```

Si creano due variabili m ed n che contengono rispettivamente il numero di righe e colonne della variabile $data$. Con la variabile M , invece, si salva il vettore dei valori medi dei segnali sulla totalità dei campioni (M è un vettore riga di 14 elementi, contenente i valori medi dei 14 segnali). A questo punto è possibile creare la variabile $data_demean$ sottraendo a $data$ un vettore costituito dai valori medi dei segnali: in tal modo si sottrae a ciascun segnale il proprio valore medio.

```

% --- Executes on button press in notchfilterbutton.
function notchfilterbutton_Callback(hObject, eventdata, handles)
    data=handles.data;
    data = double(data);
    fs=handles.fs;
    wo = 50/(fs/2);  bw = wo/45;
    [b, a] = iirnotch(wo, bw);
    data_filt = filtfilt(b, a, data);
    data=data_filt;
    handles.data=data
    guidata(hObject, handles);

```

Il codice sopra indicato realizza un filtro *notch* digitale con la “tacca” (in inglese *notch*) posta a pulsazione wo e larghezza di banda a -3dB pari a bw . L’istruzione *filtfilt()* applica un filtraggio digitale al segnale in ingresso (in questo caso $data$) senza distorsione di fase, dove i vettori b ed a , ricavati grazie all’istruzione *iirnotch()*, rappresentano rispettivamente i coefficienti al numeratore e al denominatore del filtro di secondo ordine. L’ultima istruzione è sempre l’aggiornamento della struttura $handles$ con gli ultimi puntatori aggiunti.

5.4 GENERAZIONE DEL MULTI-CHANNEL PLOT PER VISUALIZZAZIONE SIMULTANEA DEGLI ELETTRODI

Alla pressione del clic sul pushbutton *Multi-channel plot* viene richiamata l'esecuzione della funzione Callback corrispondente, di seguito riportata.

```
% --- Executes on button press in multiChannelplot.
function multiChannelplot_Callback(hObject, eventdata, handles)
    data = handles.data;
    tempi=(1:1:length(data))./handles.fs;
    offset=200;
    if handles.setnumber==0
        F4=data(:,1)+offset;
        C4=data(:,2)+2*offset;
        O2=data(:,3)+3*offset;
        T8=data(:,4)+4*offset;
        P08=data(:,5)+5*offset;
        F3=data(:,6)+6*offset;
        C3=data(:,7)+7*offset;
        O1=data(:,8)+8*offset;
        T7=data(:,9)+9*offset;
        P07=data(:,10)+10*offset;
        CZ=data(:,11)+11*offset;
        P03=data(:,12)+12*offset;
        P04=data(:,13)+13*offset;
        plot(tempi, F4, 'b', tempi, C4, 'b', tempi, O2, 'b', tempi, T8, 'b', tempi, P08, 'b', tempi, F3, 'b', tempi, C3, 'b', tempi, O1, 'b', tempi, T7, 'b', tempi, P07, 'b', tempi, CZ, 'b', tempi, P03, 'b', tempi, P04, 'b');
        yticks([mean(F4) mean(C4) mean(O2) mean(T8) mean(P08) mean(F3) mean(C3) mean(O1) mean(T7) mean(P07) mean(CZ) mean(P03) mean(P04)]);
        yticklabels({'F4', 'C4', 'O2', 'T8', 'P08', 'F3', 'C3', 'O1', 'T7', 'P07', 'CZ', 'P03', 'P04'});
    else [...]
```

Dopo aver creato il vettore dei tempi come precedentemente descritto, si crea una variabile *offset* pari a 200 utile per il distanziamento dei tracciati per una corretta visualizzazione. Infatti, ciascun segnale d'elettrodo viene incrementato di un multiplo della variabile *offset*, che rispecchia l'ordine degli elettrodi. Ad esempio, al segnale dell'elettrodo O1, cui corrisponde la colonna numero 8 di *data*, viene aggiunto un valore pari ad otto volte la variabile *offset*. Dopo aver modificato debitamente ciascun segnale, si procede alla realizzazione del grafico tramite

l'istruzione *plot* specificando le terne *ascisse*, *ordinate*, '*colore*' (ad esempio `tempi, 01, 'b'`) una di seguito all'altra. Infine, grazie alle istruzioni *yticks* e *yticklabels* si posizionano i tick dell'asse y in corrispondenza ai valori medi dei segnali (comprensivi della traslazione progressiva) e si assegnano come etichette dei tick ad le stringhe corrispondenti ai nomi dei vari elettrodi; in tal modo nel grafico, ciascun segnale mostrato è etichettato sull'asse y con il nome dell'elettrodo a cui è stato acquisito (si veda Figura 4.7) .

5.5 FILTRAGGI PASSA-BASSO E PASSA-ALTO

Il procedimento per eseguire il filtraggio passa-basso del segnale in ingresso passa attraverso tre stadi, rappresentati ciascuno dalla specifica *function*, i cui codici sono riportati nel seguito.

```
% --- Executes during object creation, after setting all properties.
function chooseLPFfreq_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject,'BackgroundColor'),
        get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end
    set(hObject,'String',{'10 Hz';'20 Hz';'30 Hz';'40 Hz';'50 Hz'});
    guidata(hObject, handles)
```

La prima funzione ad essere chiamata in gioco è la *CreateFcn* relativa al popup menu di scelta della banda passante del filtro Low Pass. Infatti, è grazie all'istruzione *set* in essa contenuta che è possibile popolare la tendina del menu di selezione con i valori delle diverse frequenze di taglio che vengono messe a disposizione all'utente per la scelta .

```
% --- Executes on selection change in chooseLPFfreq.
function chooseLPFfreq_Callback(hObject, eventdata, handles)
    items = get(hObject,'String');
    index_selected = get(hObject,'Value');
    item_selected = items{index_selected};
    display(item_selected);
    fs=handles.fs;
    Fp_LP=str2num(item_selected(1:2));
    Wp_LP=Fp_LP/(fs/2);
    Ws_LP=(Fp_LP+5)/(fs/2);
    assignin('base','Fp_LP',Fp_LP)
    handles.Wp_LP=Wp_LP;
```

```
handles.Ws_LP=Ws_LP;
guidata(hObject, handles);
```

Dopo aver eseguito la *CreateFcn*, viene eseguita la funzione di Callback corrispondente al codice di controllo da eseguire in corrispondenza alla selezione della banda passante fatta dall'utente nel popup menu. Grazie a questa funzione viene impostata la pass-band e la stop-band del filtro passa-basso. Con *Wp_LP* viene indicata la pulsazione di pass e con *Ws_LP* quella di stop. La pass-band del filtro coincide quella scelta dall'utente nel popup menu; si è deciso di impostare la stop-band del filtro a un valore pari alla pass-band più 5 Hz.

```
% --- Executes on button press in LPFbutton.
function LPFbutton_Callback(hObject, eventdata, handles)
    data = handles.data;
    data = double(data);
    Wp = handles.Wp_LP;
    Ws = handles.Ws_LP;
    Rp = 1;
    Rs = 60;
    [N, Wp] = ellipord(Wp, Ws, Rp, Rs);
    [B, A] = ellip(N, Rp, Rs, Wp, 'low');
    data_filt = filtfilt(B, A, data);
    data=data_filt;
    handles.data=data;
    guidata(hObject, handles);
```

Una volta che l'utente ha selezionato la frequenza di taglio del filtro passa basso (selezione che comporta l'esecuzione della *CreateFcn* e della funzione *Callback* appena descritte), affinché il filtraggio venga effettivamente realizzato, l'utente deve fare clic con il mouse sul pulsante LPF nella GUI. Quello sopra riportato è il codice di controllo di tale pushbutton.

Per prima cosa vengono richiamate dalla struttura *handles* le variabili *data* (poi convertita in tipo *double*), *Wp* e *Ws* (cioè la frequenza di taglio o pass-band e la frequenza che individua l'inizio della stop band) Si creano due nuove variabili *Rp* e *Rs*: la prima rappresenta il valore, in decibel, della massima perdita in banda passante; la seconda è il valore, in decibel, dell'attenuazione che subisce il segnale in banda attenuata.

Tali specifiche vengono poste in ingresso all'istruzione *ellipord()* che restituisce l'ordine più basso, N , del filtro ellittico che non perde più di R_p dB nella banda passante e ha almeno R_s dB di attenuazione nella banda di stop. Viene inoltre restituito lo scalare (o vettore) delle corrispondenti frequenze di taglio W_p .

Ora, noti N e W_p , tramite l'istruzione *ellip()* si calcolano i coefficienti A del numeratore e B del denominatore della funzione di trasferimento del filtro. Infine, grazie ad essi e per mezzo dell'istruzione *filtfilt()* si esegue il vero e proprio filtraggio.

Si noti che si è scelto di utilizzare un filtro di tipo ellittico (e non altri filtri, ad es. di Butterworth o Chebyshev) in quanto, a parità di specifiche richieste, il filtro ellittico generalmente richiede un numero minore di poli, cioè un ordine più basso.

5.6 CALCOLO DELLA DENSITÀ SPETTRALE DI POTENZA PER LE FASI DELL'ESPERIMENTO

Per computare la densità spettrale di potenza del segnale differenziata per le varie fasi di un esperimento, la prima cosa da fare è sapere quante sono le fasi in cui dividere il segnale e quanto dura ciascuna di esse. Queste funzionalità sono implementate tramite due edit-box dedicati (si veda Figura 4.10), chiamati rispettivamente *stepnumber* e *stepduration*. I codici di controllo contenuti nelle funzioni di Callback corrispondenti sono mostrati di seguito:

```
function stepnumber_Callback(hObject, eventdata, handles)
    Nfasi_PSD=str2double(get(hObject,'String'));
    handles.Nfasi_PSD=Nfasi_PSD;
    guidata(hObject, handles);
```

```
function stepduration_Callback(hObject, eventdata, handles)
    DFase_PSD=str2double(get(hObject,'String'));
    handles.DFase_PSD=DFase_PSD;
    guidata(hObject, handles);
```

grazie alla funzione *get* viene assegnato alla variabile *Nfasi_PSD* il valore digitato dall'utente per impostare il numero delle fasi dell'esperimento. Tale valore viene

salvato nella struttura *handles* ed infine, grazie alla funzione *guidata()*, viene aggiornata la struttura *handles* stessa. Funzionamento analogo per la definizione della durata, in secondi, di ciascuna fase, grazie alla variabile *DFase_PSD*.

Una volta che l'utente abbia definito tramite tastiera il numero di fasi e la rispettiva durata, per computare la PSD del segnale è necessario, come già annunciato nel Capitolo 4, impostare la lunghezza (in campioni) della finestra per il calcolo della PSD. Questa operazione, possibile grazie all'edit-box *Window* (si veda Figura 4.10), è caratterizzata dal codice Matlab sottostante:

```
function window_seconds_step_Callback(hObject, eventdata, handles)
    fs=handles.fs;
    window=get(hObject,'String');
    if window=='-',
        window=[]; %default
    else
        window=str2double(get(hObject,'String'))*fs;
    end
    handles.WFase_PSD=window;
    guidata(hObject, handles);
```

Infatti, tramite il primo comando *get* si assegna alla variabile *window* il valore, in secondi, della finestra temporale digitata dall'utente nel box dedicato; a questo punto, grazie ad un costrutto *if...else*, se l'utente non specifica alcun valore (e quindi la stringa risulta “ — ”), il valore di *window* viene impostato di default ad un valore tale per cui il segnale è diviso in otto segmenti di lunghezza uguale, mentre se egli aveva specificato un certo valore questo viene prima convertito da stringa a valore numerico (*double*) e poi moltiplicato per la frequenza di campionamento (campioni al secondo). Il risultato è quindi una variabile *window* in campioni. Le ultime due linee di codice servono per salvare la variabile *window* nella struttura *handles* ed aggiornare la struttura *handles* stessa.

Il successivo passo è impostare la risoluzione in frequenza per la PSD, il che corrisponde a specificare il numero di punti della Trasformata Discreta di Fourier (DFT) da utilizzare per la stima. Questa operazione, possibile grazie all'edit-box *Resolution* (si veda Figura 4.10), è basata sul codice Matlab sottostante:

```

function resolution_hertz_step_Callback(hObject, eventdata, handles)
    fs=handles.fs;
    nfft=get(hObject,'String');
    if nfft=='-',
        nfft=[]; %default
    else
        nfft=(1/str2double(get(hObject,'String')))*fs;
    end
    handles.nfftFase_PSD=nfft;
    guidata(hObject, handles);

```

Anche qui tramite la funzione *get* si assegna alla variabile *nfft* il valore contenuto nel campo stringa dell'edit-box dedicato, ovvero quello digitato dall'utente. A questo punto se nessun valore è stato specificato, Matlab assegna un valore di default, mentre se l'utente aveva specificato un valore, esso viene convertito in valore numerico dall'istruzione *str2double*, ne viene fatto il reciproco, e il tutto viene moltiplicato per la frequenza di campionamento *fs*. Le ultime due linee di codice servono per salvare la variabile *nfft* nella struttura *handles* ed aggiornare la struttura *handles* stessa.

Valori tipici delle variabili *window* e *nfft* sono rispettivamente 10 secondi e 0.1 Hz. Invece, per quanto riguarda il numero di fasi e la loro durata: per gli esperimenti realizzati in LIB con Neurowave l'utente deve digitare 3 fasi e 300 secondi; per gli esperimenti realizzati nel CAVE con OpenBCI l'utente deve digitare 4 fasi e 300 secondi.

A questo punto, dopo aver impostato i valori di queste quattro fondamentali variabili, l'utente è pronto a premere il pulsante azzurro per applicare l'algoritmo di calcolo della PSD. Il codice di controllo relativo alla funzione di Callback corrispondente a tale pushbutton è mostrata di seguito:

```

function applyPSDstepbystep_Callback(hObject, eventdata, handles)
    data=handles.data;
    [m,n]=size(data);
    fs=handles.fs;
    labels=handles.labels;
    window=handles.WFase_PSD;
    nfft=handles.nfftFase_PSD;
    durata=handles.DFase_PSD;
    NFasi=handles.Nfasi_PSD;
    PSD_tot=zeros(nfft/2+1, n, NFasi);
    for j=1:NFasi,

```

```

start=1+(j-1)*durata*fs;
if j~=NFasi,
    stop=j*durata*fs;
else
    stop=m;
end
[PSD, f]=pwelch(data(start:stop, :), window, [], nfft, fs);
PSD_tot(:, :, j)=PSD;
end
assignin('base', 'diversefasiPSD_tot', PSD_tot);
assignin('base', 'totalsignalFreq', f);
figure,
for i = 1:n-1
    if handles.setnumber==1
        subplot(3, 4, i)
    else
        subplot(4, 4, i)
    end
    for j=1:NFasi,
        plot(f(10:end), PSD_tot(10:end, i, j), 'linewidth', 1);
        xlabel('frequency (Hz)')
        grid on, hold on
    end
    title(labels{1, i})
    xlim([0 25]), ylim([0 20]),
    xlabel('frequency (Hz)'), ylabel('PSD (uV2/Hz)')

end
if handles.setnumber==0
    legend('pre-', 'task', 'post-')
else legend('VR off pre-', 'VR on pre-', 'task', 'VR on post-')
end
end

```

Nella prima parte del suddetto codice si richiamano, dalla struttura *handles*, le variabili necessarie alla prosecuzione dello stesso, e le si riassegna con un nome più semplice e fruibile. È questo il caso delle variabili *fs* (frequenza di campionamento dei segnali), *labels* (etichette degli elettrodi), *window* (lunghezza, espressa in campioni, in cui viene suddiviso ciascun segnale per la stima della densità di potenza), *nfft* (numero di campioni su cui viene calcolata la trasformata discreta di Fourier per la stima della densità di potenza), *durata* (durata in secondi di ciascuna fase in cui è suddiviso l'esperimento), *NFasi* (numero di fasi in cui è suddiviso l'esperimento). A questo punto, tramite l'istruzione *zeros*, si crea una matrice tridimensionale fatta di zeri, chiamata *PSD_tot*, le cui dimensioni sono: $nfft/2+1$, n ,

$NFasi$ (dove n è il numero di canali di acquisizione). Pertanto, da un punto di vista concettuale, questa variabile è costituita da $NFasi$ matrici $(nfft/2+1) \times (n)$ impilate una sull'altra.

Ora, per mezzo del costrutto *for*, l'algoritmo popola iterativamente gli elementi della matrice di zeri con i valori di PSD ottenuti dalla funzione *pwelch*, e lo fa per tutte le righe e le colonne, e soprattutto fase per fase (la terza dimensione della matrice è, come detto, indicativa della fase dell'esperimento). Si può sinteticamente spiegare come opera l'istruzione *pwelch* nel seguente modo. Con tale istruzione, il segnale di cui si vuole stimare la densità spettrale di potenza, è suddiviso in spezzoni. Dunque viene stimata la densità di potenza di ogni spezzone calcolandone la trasformata discreta di Fourier (FFT), quindi considerando il valore assoluto dei coefficienti della FFT al quadrato e dividendolo per $F_s \cdot N$ (dove F_s è la frequenza di campionamento del segnale e N è il numero di campioni di ogni spezzone). Quindi le densità di potenza calcolate sui vari spezzoni vengono mediate, e il risultato è la stima della PSD del segnale. La suddivisione in spezzoni e il calcolo del valore medio serve a ridurre la varianza della stima. La risoluzione in frequenza della densità stimata dipende dunque della lunghezza (in campioni) di ciascun spezzone ed è pari a F_s/N (dove N è il numero di campioni di ciascun spezzone). Ora, per avere bassa varianza sarebbe opportuno suddividere il segnale in un numero elevato di spezzoni (così da mediare su tanti spezzoni), ma questo vorrebbe dire avere spezzoni con un numero più basso di campioni N e quindi avere una bassa risoluzione in frequenza. Si ha dunque un compromesso tra varianza e risoluzione. Per migliorare questo compromesso, l'istruzione *pwelch* mette a disposizione la possibilità di specificare separatamente il numero di campioni di ciascun spezzone e il numero di punti su cui calcolare la *fft*. Se il numero di punti su cui calcolare la *fft* è maggiore del numero di campioni di ogni spezzone, l'istruzione *pwelch* esegue uno zero-padding, ovvero aggiunge degli zero in fondo a ciascun spezzone su cui calcola la *fft*. In questo modo non viene alterato il contenuto spettrale di ogni spezzone, ma ne viene aumentato il numero di campioni (su cui viene calcolata la *fft*) e così è possibile migliorare il compromesso varianza-risoluzione (avere bassa varianza con un numero elevato di spezzoni e buona risoluzione, cioè *fft* calcolata su più campioni). La sintassi dell'istruzione *pwelch* è la seguente: $[PSD, f]=pwelch(X,window,[],nfft,fs)$. X (primo argomento in

ingresso) è il segnale di cui si vuole stimare la densità di potenza. *window* (secondo argomento in ingresso) specifica il numero di campioni di ciascun spezzone in cui viene suddiviso il segnale. Ci può essere una sovrapposizione (overlap) tra spezzoni adiacenti di un numero di campioni pari al valore contenuto nel terzo argomento di ingresso; nel caso specifico, il terzo argomento è vuoto (*[]*) e viene dunque usato un valore di overlap di default pari al 50%, cioè il segnale è suddiviso in spezzoni ognuno contenente un numero di campioni pari al valore di *window* e che si sovrappongono per il 50% dei campioni). *nfft* (quarto argomento in ingresso) è il numero di punti su cui viene calcolata la FFT per ogni spezzone; quindi, come detto precedentemente se tale argomento ha valore superiore a *window* viene eseguito uno zero-padding. Infine, l'ultimo argomento (*fs*) specifica la frequenza di campionamento del segnale *X* e serve a restituire come argomenti di uscita, oltre ai valori della densità di potenza PSD (primo argomento di uscita), il vettore delle frequenze *f* (secondo argomento di uscita) in corrispondenza ai quali sono calcolati i valori di PSD ($f = [0:fs/nfft: fs/2-1/nfft]$). Se *X* è una matrice (come nel caso in esame), l'istruzione *pwelch* calcola densità di potenza su ogni colonna di *X* e restituisce i valori nella colonna corrispondente dell'argomento di uscita *PSD*.

Una volta calcolata la densità di potenza dei vari canali e per ciascuna fase e salvati i valori nella variabile *PSD_tot* tramite la funzione *assignin* si salvano le variabili *PSD_tot* e *f* nel Workspace di Matlab con i nomi, rispettivamente, 'diversefasiPSD_tot' e 'totalsignalFreq'. In questo modo, senza "appesantire" ulteriormente la struttura dei puntatori handles, si sono rese fruibili agli altri object della GUI queste variabili.

Una volta conclusa la computazione dei valori di PSD, si passa alla loro rappresentazione grafica. Infatti, viene generata una nuova figura e la si suddivide, per mezzo del comando *subplot*, in 12 o 13 sotto-figure in base al sistema di acquisizione utilizzato. A questo punto, su ciascuna di queste sotto-figure viene generato il grafico della densità spettrale di potenza relativa al segnale di quell'elettrodo, differenziata nelle diverse fasi dell'esperimento. In seguito, vengono aggiunte etichette dei nomi sugli assi, titolo delle figure, e legenda corrispondente.

5.7 CALCOLO DELLA POTENZA IN UNA SPECIFICA BANDA DEI SEGNALI EEG E SALVATAGGIO DEI RISULTATI

Dopo aver calcolato e rappresentato la PSD, l'utente può proseguire ed ultimare l'analisi in frequenza dei segnali EEG selezionati computando la potenza di tali segnali in specifiche bande di frequenza. Nella GUI sono stati progettati due pushbutton che implementano il suddetto calcolo nelle bande alpha e beta. Nel seguito si riporta e si commenta il codice di controllo della funzione Callback del primo di questi, chiamato *alphapowersave*.

```
% --- Executes on button press in alphapowersave.
function alphapowersave_Callback(hObject, eventdata, handles)
    f=evalin('base','totalsignalFreq');
    PSD=evalin('base','diversefasiPSD_tot');
    iniz=find(f==8);
    fin=find(f==14);
    flim=f(iniz:fin);
    i=1;
    fine=39-3*handles.setnumber;
    myfigure= figure
    for j=1:3:fine
        PSD_experiment_f1=PSD(iniz:fin, i, 1);
        PSD_experiment_f2=PSD(iniz:fin, i, 2);
        PSD_experiment_f3=PSD(iniz:fin, i, 3+handles.setnumber);
        apower_experiment_f1=trapz(flim, PSD_experiment_f1);
        apower_experiment_f2=trapz(flim, PSD_experiment_f2);
        apower_experiment_f3=trapz(flim, PSD_experiment_f3);
        apower_vector(j) = apower_experiment_f1;
        apower_vector(j+1)= apower_experiment_f2;
        apower_vector(j+2)= apower_experiment_f3;
        subplot(4-handles.setnumber, 4, i),
        plot(apower_vector(j:j+2), 'b*')
        title(handles.labels{1, i})
        xlim([0.5 3.5])
        set(gca, 'xtick', [1 2 3])
        if handles.setnumber==0
            set(gca, 'xticklabel', {'pre', 'task', 'post'})
        else set(gca, 'xticklabel', {'r1', 'r1VR', 'r2VR'})
        end
        grid on, grid minor
        i=i+1;
    end
    nome=evalin('base','stringa_nome_soggetto')
    if handles.setnumber==0
        nome_salvataggio=[nome(7:11), '_apower'];
```

```

else nome_salvataggio=[nome(9:16), '_apower'];
end
assignin('base', nome_salvataggio, apower_vector
experiment = evalin('base', 'chosen_experiment');
if handles.setnumber==0
    save(['C:\Users\Gianlu\Documents\tesi\segnali
Neurowave\risultati analisi\' , nome_salvataggio], 'apower_vector')
else
    save(['C:\Users\Gianlu\Documents\tesi\segnali
OpenBCI\risultati analisi\' , nome_salvataggio], 'apower_vector')
end
end

```

Dopo aver caricato dal Workspace le variabili f ('totalsignalFreq') e PSD ('diverseFasiPSD_tot') grazie all'istruzione *evalin()*, vengono individuati, nel vettore delle frequenze, gli indici *iniz* e *fin* a cui corrispondono gli estremi della banda *alpha*, ovvero 8 Hz e 14 Hz. Solo ora è possibile estrarre dall'intero vettore delle frequenze l'intervallo di integrazione di interesse, e assegnarlo come variabile *flim*. A questo punto si definisce la variabile *fine* che assume il valore 39 (13*3, ovvero 13 valori di potenza in banda alpha, uno per ogni elettrodo, in ciascuna delle tre fasi), se è selezionato Neurowave (setnumber=0), oppure 36 (12*3, ovvero 12 valori di potenza in banda alpha, uno per ogni elettrodo, in ciascuna delle tre fasi), se è selezionato OpenBCI (setnumber=1). Ora si accede al ciclo for: per prima cosa vengono create tre nuove variabili rappresentative rispettivamente della prima, seconda e terza fase dell'esperimento in questione. Notare che, grazie all'utilizzo della variabile *setnumber*, la "fase 3" indicata come *PSD_experiment_f3*, corrisponde alla effettiva fase 3 degli esperimenti Neurowave (setnumber=0), ma alla fase 4 degli esperimenti OpenBCI (setnumber=1), permettendo di "saltare" la fase di interazione con la Virtual Reality, in cui i segnali sono poco affidabili a causa dei movimenti del soggetto, e considerare al suo posto la fase di relax finale a VR accesa.

Per calcolare la potenza dei segnali in banda *alpha*, l'algoritmo si serve della funzione *trapz*, che realizza la cosiddetta *integrazione numerica trapezoidale*, che computa l'area sottesa ad una curva approssimandone il valore con il metodo dei trapezi.

Questo metodo approssima l'integrazione su un intervallo suddividendo l'area in trapezi con area più facilmente calcolabile. Per una integrazione con N+1 punti equamente spazati, l'approssimazione è data dalla formula:

$$\int_a^b f(x) dx \approx \frac{b-a}{2N} \sum_{n=1}^N (f(x_n) + f(x_{n+1}))$$

$$= \frac{b-a}{2N} [f(x_1) + 2f(x_2) + \dots + 2f(x_N) + f(x_{N+1})]$$

dove il distanziamento fra ciascun punto è pari allo scalare $\frac{b-a}{N}$.

I tre valori ottenuti dalle funzioni *trapz* in queste tre linee di codice vengono assegnati a tre elementi consecutivi di un nuovo vettore, chiamato *apower_vector*. Questo vettore è costituito o da 12 o da 13 terne di valori corrispondenti ai valori di potenza *alpha* calcolati dall'algoritmo per ciascuna fase per ciascun elettrodo d'acquisizione EEG. Pertanto, se era stato selezionato Neurowave (13 elettrodi EEG) il vettore conterrà 39 elementi, mentre se la scelta era ricaduta su OpenBCI (12 elettrodi EEG) tale vettore ne conterrà 36. In tale vettore gli elementi sono organizzati ed ordinati nel seguente modo: potenza alpha dell'elettrodo 1 in fase 1, potenza alpha dell'elettrodo 1 in fase 2, potenza alpha dell'elettrodo 1 in fase 3, potenza alpha dell'elettrodo 2 in fase 1, potenza alpha dell'elettrodo 2 in fase 2, e così via, procedendo per terne di valori, per tutti gli elettrodi.

Sempre all'interno del ciclo *for*, si generano tanti *subplot* quanti sono gli elettrodi in questione, e in ciascuno di essi si rappresenta, per mezzo di tre asterischi blu, il valore di potenza alpha nelle tre fasi dell'esperimento (si veda Figura 4.13), si assegna un titolo con l'etichetta dell'elettrodo corrispondente, e si infittisce la griglia. Infine, l'istruzione *i=i+1* chiude il ciclo *for*: è in questo modo che l'indice *i*, rappresentativo dell'elettrodo, viene incrementato e permette di scorrere tutti gli elettrodi.

Concluso l'algoritmo iterativo del ciclo *for*, è la volta del codice per la generazione del nome di salvataggio dei risultati ed il codice di salvataggio stesso. Brevemente, il primo si basa sul concetto di estrarre dal nome del file caricato una stringa identificativa del soggetto e dell'esperimento e quindi di concatenarla alla desinenza '*_apower*', il secondo di salvare il vettore delle potenze suddetto nella cartella appropriata. Ciò che si ottiene è, ad esempio, un file chiamato "SS0_c_apower.mat" nella cartella *Risultati analisi* di Neurowave ed un file chiamato "SOGG4_B2_apower.mat" nella cartella *Risultati analisi* di OpenBCI.

Il codice di controllo della funzione di Callback di *alphapowersave* continua con il calcolo della potenza media, in banda *alpha*, per regione dello scalpo, ovvero computando una media aritmetica dei valori ottenuti da gruppi di elettrodi. In particolare, sono state create cinque famiglie di elettrodi: *frontali*, *centrali*, *temporali*, *occipitali* e *parieto-occipitali*. In questo modo è possibile ottenere una rappresentazione più sintetica della distribuzione dell'alpha, per macroregioni di interesse anziché per singoli elettrodi.

```

a=apower_vector;
if handles.setnumber==0
    amean_frontali=[mean([a(1) a(16)]) mean([a(2) a(17)])
mean([a(3) a(18)])];
    amean_centrali = [mean([a(4) a(19) a(31)]) mean([a(5) a(20)
a(32)]) mean([a(6) a(21) a(33)])];
    amean_occipitali =[mean([a(7) a(22)]) mean([a(8) a(23)])
mean([a(9) a(24)])];
    amean_temporali = [mean([a(10) a(25)]) mean([a(11) a(26)])
mean([a(12) a(27)])];
    amean_parietooccipitali =[mean([a(13) a(28) a(34) a(37)])
mean([a(14) a(29) a(35) a(38)]) mean([a(15) a(30) a(36) a(39)])];
else
    amean_frontali = [mean([a(1) a(19)]) mean([a(2) a(20)])
mean([a(3) a(21)])];
    amean_centrali = [mean([a(4) a(22)]) mean([a(5) a(23)])
mean([a(6) a(24)])];
    amean_occipitali =[mean([a(16) a(34)]) mean([a(17)
a(35)]) mean([a(18) a(36)])];
    amean_temporali = [mean([a(7) a(25)]) mean([a(8) a(26)])
mean([a(9) a(27)])];
    amean_parietooccipitali =[mean([a(10) a(13) a(28)
a(31)]) mean([a(11) a(14) a(29) a(32)]) mean([a(12) a(15)
a(30) a(33)])];
end

```

Infine, dopo aver calcolato i cinque valori di potenze *alpha* medie per ciascuna delle tre fasi (per un totale di 15 valori), si realizza un grafico molto importante, con il quale è possibile visualizzare la variabilità, regione per regione, della potenza *alpha* registrata nelle tre fasi:

```

figure,
plot(1:3,amean_frontali,'k',4:6,amean_centrali,'r',7:9,amean_temporali,'
b',10:12,amean_parietooccipitali,'m',13:15,amean_occipitali,'g')
legend('FRONTAL','CENTRAL','TEMPORAL','PARIETO-OCCIPITAL','OCCIPITAL')
title('Average alpha power per region'),
xticks([]),
ylabel('Power (uV^2)')

```

Tutti i codici e le relative spiegazioni fornite nel presente sotto-capitolo riguardano, come già detto, la trattazione della banda *alpha*. Si precisa che alla pressione sul pushbutton Beta Power si attiva la funzione di Callback *betapowersave*, il cui codice (e quindi funzionamento) è strutturalmente identico a quello per *alpha*. L'unica differenza riguarda gli estremi della banda: da 14 Hz a 25 Hz, dal momento che questo è l'intervallo di frequenze proprie della banda beta.

5.8 VISUALIZZAZIONE DEL SEGNALE ECG ED ANALISI DELL'HRV

Come illustrato nel precedente capitolo, la GUI che ho realizzato offre all'utente la possibilità di condurre anche un'analisi elettrocardiografica dei segnali registrati durante l'esperimento: visualizzazione ECG nel tempo; visualizzazione del Tacogramma; calcolo e rappresentazione della frequenza cardiaca (HR); computazione della PSD del segnale nelle diverse fasi dell'esperimento, calcolo delle potenze LF ed HF, ed infine calcolo del rapporto LF/HF. Dal momento che le funzioni eseguite in questa sezione (filtraggio, stima della densità di potenza, calcolo della potenza in LF e HF) usano le stesse istruzioni Matlab già viste per i segnali EEG (*ellipord*, *ellip*, *filtfilt*, *pwelch*, *trapz*, etc.), si ritiene opportuna una trattazione più sintetica del codice corrispondente.

Il pushbutton che implementa la visualizzazione dell'ECG nel tempo presenta la seguente funzione di Callback:

```
% --- Executes on button press in Click2ECGanalysis.
function Click2ECGanalysis_Callback(hObject, eventdata, handles)
    ecgsignal=handles.data(:,end);
    figure
    tempi=(1:length(ecgsignal))/handles.fs;
    plot(tempi, ecgsignal,'b')
    title('Electrocardiogram (ECG)'), xlabel('Time (s)'),
    ylabel('Amplitude (uV)'), grid on, grid minor
    xlim([0 length(ecgsignal)/handles.fs])
```

Il pushbutton che implementa la visualizzazione del Tacogramma del segnale presenta la seguente funzione di Callback:

```
% --- Executes on button press in Tachogram.
function Tachogram_Callback(hObject, eventdata, handles)
    nome_per_IBI=handles.nome_per_IBI;
    load(['C:\Users\Gianlu\Documents\tesi\segnali
```

```

Neurowave\IBIs\NWIBIs_', nome_per_IBI])
RR=IBIs/1000;
assignin('base','RR',RR)
C=cumsum(RR);
t=[0;C(1:end-1)];
assignin('base','t',t)
figure, subplot(211)
plot(t,RR,'linewidth',1)
ylim([0.3 1.3]), title('Tachogram-RR'),ylabel('s'),xlabel('s'),
grid on, grid minor
subplot(212)
plot([1:1:length(RR)],RR,'linewidth',1)
ylim([0.3 1.3]), xlim([0 length(RR)]),title('Tachogram-RR')
ylabel('s'), xlabel('#beat'), grid on, grid minor
guidata(hObject, handles);

```

Il pushbutton HRbutton calcola la frequenza cardiaca, la rappresenta nel tempo, la filtra e la interpola linearmente per poi rappresentarla di nuovo. Il codice di controllo corrispondente è:

```

% --- Executes on button press in HRbutton.
function HRbutton_Callback(hObject, eventdata, handles)
    RR=evalin('base','RR');
    HR=1./RR;
    assignin('base','HR',HR)
    figure,subplot(211)
    t=evalin('base','t');
    plot(t,HR,'linewidth',1)
    ylim([0.5 2]),title('Heart rate '),ylabel('bps'), xlabel('s')
    Fc_interp=10; %Hz
    assignin('base','Fc_interp',Fc_interp)
    Tc_interp=1/Fc_interp;
    t_interp=[t(1):Tc_interp:t(end)];
    RR_interp=interp1(t,RR,t_interp);
    HR_interp=1./RR_interp;
    [N, Wp] = ellipord(0.5/(Fc_interp/2), 0.65/(Fc_interp/2), 1, 60);
    [B,A] = ellip(N,1,60,Wp);
    HR_interp_filt = filtfilt(B,A,HR_interp);
    assignin('base','HR_interp_filt',HR_interp_filt)
    subplot(212)
    plot(t_interp,HR_interp_filt,'linewidth',1)
    ylim([0.5 2])
    title('Interpolated and Filtered Heart rate')
    ylabel('bps'), xlabel('s')
    guidata(hObject, handles);

```

Infine, si riporta nel seguito il codice di controllo del pushbutton *HRVbutton*:

```
% --- Executes on button press in HRVbutton.
function HRVbutton_Callback(hObject, eventdata, handles)
    HRnew=evalin('base','HR_interp_filt');
    Fc_interp=evalin('base','Fc_interp');
    window=100*Fc_interp;
    NFFT=5000;
    [P_r1,f]=pwelch(HRnew(1:round(5*end/15)),window,[],NFFT,Fc_interp);
    [P_t,f]=pwelch(HRnew(round(5*end/15):round(10*end/15)),window,[],NFFT,Fc_interp);

    [P_r2,f]=pwelch(HRnew(round(10*end/15):end),window,[],NFFT,Fc_interp);
    iniz=max(find(f<=0.02));
    fin=min(find(f>=0.5));
    figure
    plot(f(iniz:fin),P_r1(iniz:fin),'linewidth',1.5)
    xlim([0.0 0.5]),hold on
    plot(f(iniz:fin),P_t(iniz:fin),'linewidth',1.5)
    plot(f(iniz:fin),P_r2(iniz:fin),'linewidth',1.5)
    legend('rest-pre','task','rest-post')
    title('PSD (s2/Hz)'), xlabel('Frequency (Hz)')
    min_area_LF=min(find(f>=0.04));
    max_area_LF=max(find(f<=0.15));
    min_area_HF=min(find(f>=0.15));
    max_area_HF=max(find(f<=0.4));
    area_pre_HR_LF=trapz(P_r1(min_area_LF:max_area_LF))*0.002;
    area_pre_HR_HF=trapz(P_r1(min_area_HF:max_area_HF))*0.002;
    area_task_HR_LF=trapz(P_t(min_area_LF:max_area_LF))*0.002;
    area_task_HR_HF=trapz(P_t(min_area_HF:max_area_HF))*0.002;
    area_post_HR_LF=trapz(P_r2(min_area_LF:max_area_LF))*0.002;
    area_post_HR_HF=trapz(P_r2(min_area_HF:max_area_HF))*0.002;
    HR_LFHFpower_vector=[area_pre_HR_LF area_pre_HR_HF area_task_HR_LF
    area_task_HR_HF area_post_HR_LF area_post_HR_HF]
    save(['C:\Users\Gianlu\Documents\tesi\segnali Neurowave\risultati
    analisi\HR_LFHFpw_vect_',handles. nome_per_IBI],'HR_LFHFpower_vector')
    LFHFratio_pre=area_pre_HR_LF/area_pre_HR_HF
    LFHFratio_task=area_task_HR_LF/area_task_HR_HF
    LFHFratio_post=area_post_HR_LF/area_post_HR_HF
```

CONCLUSIONI

In questo lavoro di tesi è stata sviluppata una Interfaccia Grafica Utente (GUI) in MATLAB[®] testata ed applicata su segnali EEG ed ECG acquisiti su volontari sani in due diversi esperimenti svolti presso i laboratori dell'Università di Bologna da docenti dei Dipartimenti DEI (raggruppamento di Bioingegneria) e DIN, al fine di consentire la visualizzazione e l'analisi spettrale di questi segnali mediante una grafica user-friendly. Uno dei due esperimenti era finalizzato a valutare l'effetto sui soggetti di diverse componenti attentive; l'altro esperimento era finalizzato a investigare l'effetto di ambienti di realtà virtuale. Si ricorda che l'attenzione è stata posta sui potenziali bioelettrici EEG ed ECG poiché essi sono segnali spontanei, facili da acquisire in modo non invasivo, e soprattutto perché sono soggetti a modificazioni in risposta a variazioni dello stato attentivo, di stress, di gradimento e di coinvolgimento del soggetto. Nei mesi precedenti alla realizzazione del programma in Matlab vero e proprio, ho partecipato attivamente alle acquisizioni su alcuni dei soggetti coinvolti in questi esperimenti.

Nel presente elaborato, prima di illustrare nel dettaglio la specifica Interfaccia Utente realizzata, sono stati illustrati i principi di base della realizzazione di una GUI in Matlab, sia dal punto di vista dell'utente (livello front-end) che del programmatore (livello del codice). Infatti, tramite due semplici interfacce grafiche realizzate *ad hoc*, il lettore è introdotto (nel Capitolo 3), agli *object* (elementi fondamentali per la realizzazione di una Interfaccia Grafica), alle loro proprietà e ai codici di controllo. In questo modo, si rende più agevole la comprensione dei successivi contenuti, relativi alle funzionalità della specifica GUI realizzata (Capitolo 4) e del sottostante codice di implementazione (Capitolo 5).

Nel seguito vengono illustrati i principali pregi e limiti della GUI, i possibili sviluppi futuri e le applicazioni.

Un importante aspetto positivo dell'interfaccia è che è user-friendly: il suo funzionamento è facilmente comprensibile e gli step che l'utilizzatore deve seguire sono suggeriti dal raggruppamento delle diverse funzioni in pannelli di colore differente. Si è cercato, infatti, da questo punto di vista, di curare l'aspetto estetico della GUI.

Un altro aspetto di pregio è la libertà lasciata all'utente nella scelta dei valori di alcuni parametri, conferendo flessibilità di utilizzo alla GUI. Un primo esempio è l'inserimento della frequenza di campionamento. Naturalmente, il valore della frequenza di campionamento va inserito correttamente sulla base di quello adottato dal sistema di acquisizione e conversione analogica/digitale dei segnali, per ottenere una visualizzazione realistica dei segnali e risultati dell'analisi affidabili. Ora, mentre per gestire correttamente i dati acquisiti dai due sistemi di acquisizione utilizzati negli esperimenti occorre indicare una precisa frequenza di campionamento (128 Hz e 125 Hz nei due casi) si è deciso di non preimpostare tale valore ma di renderlo inseribile ogni volta dall'utente. In questo modo, in futuro la GUI potrà essere adattata e usata anche per gestire dati acquisiti a frequenza diverse. Altri esempi di parametri impostabili dall'utente sono rappresentati dai parametri che controllano l'analisi spettrale e il calcolo della densità spettrale di potenza dei segnali, attraverso specifici edit-box ("Step n°", "Step duration", "Window", "Resolution") contenuti nel pannello Frequency Domain della GUI. I primi due parametri, consentono di suddividere gli interi tracciati (ottenuti sull'intera durata dell'esperimento) in un certo numero di porzioni separate ("Step n°"), specificando la durata di tali porzioni ("Step duration"). Questo permette di fare l'analisi spettrale separatamente su fasi diverse dell'esperimento (ad es. fase di relax e fase di task). Infatti, sebbene in relazione ai due specifici esperimenti abbia principalmente senso la suddivisione in tre fasi di ciascun esperimento, si lascia comunque flessibilità dell'analisi permettendo all'utente di fare anche suddivisioni diverse degli interi tracciati. Inoltre, e ancora più importante, in futuro la GUI potrebbe essere utilizzata per segnali acquisiti in esperimenti totalmente diversi dai due qui considerati, per i quali l'utente potrà autonomamente decidere la suddivisione più appropriata. Gli altri parametri ("Window" e "Resolution") sono parametri specificamente legati al metodo di stima della densità spettrale di potenza e consentono di modificare la durata ("Window") delle finestre temporali in cui viene suddiviso il segnale (intero o una sua porzione sulla base della ripartizione in fasi) per la stima della sua densità di potenza e la risoluzione in frequenza ("Resolution") dello spettro di densità di potenza. Infatti, la scelta di questi parametri può anche essere influenzata da vari aspetti quali la durata del segnale sotto analisi o le frequenze che si vogliono

analizzare. Una flessibilità in questo senso è utile, quindi, anche per un utilizzo futuro della GUI su segnali acquisiti in esperimenti diversi.

Un altro punto di forza è che l'interfaccia può essere facilmente modificata e ampliata per introdurre nuove funzionalità. A tal proposito si ricorda che la GUI dispone di un pannello libero - detto *Future uses Panel* - entro il quale implementare nuove funzionalità in risposta a nuove esigenze che potrebbero nascere in futuro. Infatti, uno degli sviluppi di questa GUI potrebbe essere un pannello dedicato all'analisi statistica (si veda anche in seguito). Il carattere modulare della GUI è rappresentato anche da un altro piccolo esempio: il pushbutton “--- Power” presente nella sezione *Compute and Store signal Power* del *Frequency Domain* permette di aggiungere la funzionalità di calcolo della potenza del segnale EEG in altre bande di interesse, oltre al calcolo della potenza nelle bande *alpha* (8-14 Hz) e *beta* (14-25 Hz) già implementate.

Ovviamente, oltre agli aspetti positivi sopra sottolineati, il lavoro presenta alcuni limiti.

Un primo limite è relativo al fatto che la GUI non prevede la possibilità di rimuovere artefatti dai segnali EEG. Il segnale EEG è infatti fortemente contaminato da segnali di natura non cerebrale quali segnali di origine oculare, cardiaca o muscolare, che alterano il segnale stesso e che possono quindi influire sui risultati dell'analisi. La rimozione degli artefatti dal segnale EEG (senza alterare il segnale utile) è chiaramente un problema molto complesso per il quale sono stati proposti e sono disponibili diversi metodi. Una futura espansione della GUI potrebbe prevedere di inserire una funzionalità che implementa algoritmi di rimozione di artefatti, oppure che permetta di richiamare delle funzioni Matlab già disponibili in appositi toolbox open source (ad es. EEGLAB) che eseguono la rimozione di artefatti.

Un ulteriore limite riguarda il fatto che, per quanto riguarda il segnale ECG e di frequenza cardiaca, la GUI richiede in ingresso già il tacogramma (cioè la durata dei battiti cardiaci) e non implementa la funzionalità di estrazione del tacogramma dal segnale ECG (attraverso il riconoscimento dei picchi R). Anche in questo caso, esistono e sono disponibili metodi e algoritmi per l'estrazione del tacogramma dall'ECG. Uno sviluppo futuro della GUI potrebbe prevedere di aggiungere tale

funzionalità, implementando direttamente uno di questi algoritmi o – in alternativa – richiamando l'esecuzione di un software (anche stand-alone, al di fuori dell'ambiente Matlab) che svolga tale estrazione e che importi poi in Matlab l'output (il tacogramma appunto) ottenuto attraverso di esso.

Un altro limite è relativo all'assenza di funzionalità di analisi statistica sui gruppi di soggetti acquisiti. In prospettiva, si potrebbe dotare l'interfaccia grafica di una serie di pushbutton che calcolino alcuni dei più significativi parametri di posizione o localizzazione (moda, media e mediana) e di dispersione (varianza e deviazione standard). L'analisi potrebbe essere ulteriormente arricchita, sviluppando alcuni *object* per eseguire inferenza statistica tramite la stima per intervalli di confidenza o il test d'ipotesi. In questo modo si potrebbero fare test statistici di alcune misure, quali la potenza in banda alpha o beta, sui gruppi di soggetti acquisiti analizzando eventuali differenze statisticamente significative tra le condizioni di riposo e condizioni di stimolazione/task, o in generale tra le varie fasi in cui è suddiviso l'esperimento.

L'interfaccia grafica è certamente di utilità di per sé ai ricercatori e futuri laureandi coinvolti in questo tipo di studi sperimentali come strumento rapido e intuitivo, sia per la visualizzazione dei segnali e, cosa più importante, per la loro analisi spettrale. Oltre a ciò, l'interfaccia grafica sviluppata può sicuramente avere utili finalità didattiche. Infatti, la presente GUI ben si presta ad essere utilizzata da docenti universitari, in particolare nell'ambito bioingegneristico, per illustrare aspetti relativi alla visualizzazione e all'elaborazione di segnali biomedici, sia attraverso lezioni teoriche in aula (proiettando a video la GUI e i risultati ottenuti attraverso le sue funzionalità) sia in sessioni di laboratorio (facendo autonomamente utilizzare la GUI agli studenti). In questo modo gli studenti possono prendere confidenza sui seguenti aspetti: come avviene la visualizzazione dei segnali EEG nel dominio dei tempi, come essi si modificano quando si opera un filtraggio passa-basso e/o passa-alto, cosa comporta modificare i parametri (frequenza di taglio) del filtraggio nel segnale filtrato, cosa significa calcolare la densità spettrale di potenza di un segnale, come si altera lo spettro di densità di potenza modificando i parametri nell'algoritmo di stima, come si computa la potenza dei segnali EEG in una specifica banda di frequenza. Inoltre, per mezzo della GUI, gli studenti possono

visualizzare un segnale ECG reale, visualizzare il tacogramma e la frequenza cardiaca e comprendere cosa si intende per variabilità della frequenza cardiaca.

In conclusione, dunque, l'Interfaccia Grafica realizzata è utile non solo per l'analisi dei dati acquisiti negli esperimenti effettuati ma potrà essere utilizzata anche in esperienze di laboratorio future, fornendo gli strumenti per una appropriata visualizzazione e analisi dei risultati, e anche in future esperienze didattiche, come utile mezzo di apprendimento per gli studenti. L'utilizzo del programma da parte dei futuri utilizzatori potrà anche mettere in luce quali ulteriori aspetti sono da migliorare, da rivedere o da ampliare oltre ai limiti già precedentemente evidenziati.

INDICE DELLE FIGURE E DELLE TABELLE

Figura 1.1. Tipici tracciati di un elettroencefalogramma di superficie.....	11
Figura 1.2. Vista laterale dell'encefalo: suddivisione dei 4 lobi principali	13
Figura 1.3. Suddivisione in strati della corteccia cerebrale	14
Figura 1.4. Esempi di sinapsi eccitatoria nelle regioni basali e apicali del dendrite di un neurone piramidale	17
Figura 1.5. Misurazione dei dipoli tangenziali e radiali	19
Figura 1.6. Esempi di alcuni spettri di potenza che mostrano picchi alle frequenze EEG canoniche	23
Figura 1.7. Depolarizzazione e ripolarizzazione	26
Figura 1.8. Esempio di tracciato ECG in derivazione D1	27
Figura 1.9. Heart Rate Variability ed ECG	29
Figura 1.10. ECG e tacogramma	29
Figura 1.11. Spettro di potenza del segnale HRV	33
Figura 2.1. Laboratorio CAVE	35
Figura 2.2. Tre proiettori di ultima generazione	35
Figura 2.3. Ambientazioni virtuali di cabine passeggeri: cabina B2 e cabina B1	37
Figura 2.4. Uno dei soggetti volontari durante l'esperimento OpenBCI	38
Figura 2.5. Componenti hardware del sistema OpenBCI utilizzato	43
Figura 2.6. Configurazione elettrodi e sistema Cyton+Daisy	45
Figura 2.7. Uno dei soggetti volontari durante l'esperimento Neurowave	47
Figura 2.8. Uno dei soggetti volontari e mappa di distribuzione degli elettrodi	51
Figura 2.9. Sistema di amplificazione BRAINBOX® EEG-1166	53
Figura 2.10. Parte della strumentazione utilizzata per gli esperimenti Neurowave	54
Figura 2.11. Rappresentazione del meccanismo di conduzione del segnale EEG dal cervello all'elettrodo	55
Figura 2.12. Gel conduttore OneGel della EB Neuro	56
Figura 2.13. Siringa a becco smusso	56
Figura 3.1. GUIDE Quick Start	59
Figura 3.2. Layout Editor di GUIDE	59
Figura 3.3. Palette degli uicontrol objects	61
Figura 3.4. Una semplice GUI e le proprietà di default dei suoi oggetti	64
Figura 3.5. Una semplice GUI e le proprietà modificate dei suoi oggetti	66
Figura 3.6. GUI "codificacolor"	70
Figura 3.7. Codice di inizializzazione della GUI "codificacolor"	71

Figura 3.8. Codici di apertura ed output della GUI “codificacolore”	71
Figura 3.9. Codice di controllo del pushbutton1	71
Figura 3.10. Esempi di funzionamento della GUI “codificacolore”	73
Figura 4.1. Visione d’insieme della Graphical User Interface realizzata	75
Figura 4.2. Finestra di dialogo per la scelta dell’esperimento Neurowave	78
Figura 4.3. Finestra esplora-risorse che compare alla pressione del pulsante Reading in “sceltaesperimentoNW”	79
Figura 4.4. Finestra di dialogo per la scelta dell’esperimento OpenBCI	79
Figura 4.5. Visualizzazione del segnale EEG registrato dal singolo elettrodo PO7 durante esperimento completo di Neurowave	81
Figura 4.6. Multi-channel Plot	82
Figura 4.7. Multi-channel Plot De-mean	83
Figura 4.8. Risposta in frequenza di un filtro Notch a 50 Hz	84
Figura 4.9. Densità spettrale di potenza dei segnali EEG registrati dagli elettrodi durante l’esperimento lettura Neurowave	86
Figura 4.10. Pannello di inserimento dei valori per PDS differenziata per le varie fasi dell’esperimento	87
Figura 4.11. Densità spettrale di potenza dei segnali EEG registrati dagli elettrodi durante esperimento lettura Neurowave differenziata per le tre fasi	87
Figura 4.12. Pannello per il calcolo e salvataggio della potenza del segnale in banda alpha e beta	88
Figura 4.13. Rappresentazione grafica della potenza alpha registrata da ciascun elettrodo nelle tre fasi dell’esperimento “lettura” di Neurowave	89
Figura 4.14. Curve di variabilità, nelle tre fasi dell’esperimento, della potenza alpha media per regione dello scalpo, esperimento lettura	90
Figura 4.15. Rappresentazione grafica della potenza alpha registrata da ciascun elettrodo in tre fasi dell’esperimento “cabina B2” di OpenBCI	91
Figura 4.16. Curve di variabilità, in tre fasi dell’esperimento, della potenza alpha media per regione dello scalpo, esperimento cabina B2	91
Figura 4.17. Serie temporale del segnale ECG e suo ingrandimento	92
Figura 4.18. Tacogramma in funzione del tempo e del numero di battiti.....	94
Figura 4.19. Grafico della frequenza cardiaca su tempo e della interpolata e filtrata	95
Figura 4.20. Densità spettrale di potenza del segnale di Heart Rate interpolato e filtrato.....	96
Tabella 1.1 Principali caratteristiche dei ritmi cerebrali, stato mentale a cui sono associati e possibili task per valutarne le variazioni.	19
Tabella 2.1. Elenco canali-elettrodi utilizzati con sistema OpenBCI	44
Tabella 2.2. Elenco canale-elettrodo relativa alle acquisizioni Neurowave in LIB	52

BIBLIOGRAFIA E FONTI

- [1] Avanzolini G, Magosso E (2015) “Strumentazione Biomedica. Progetto e impegno dei sistemi di misura”, Patron Bologna: pp.215-216.245-247.259-264.276-277
- [2] Olejniczak P (2006). “Neurophysiologic Basis of EEG” *Journal of Clinical Neurophysiology*. 23(3), pp.186-187
- [3] <https://brainwaves.me/tag/eeg/> consultato il 12/08/18
- [4] Silverthorn D. U., (2014) “Fisiologia umana. Un approccio integrato”, Pearson Italia – Milano, Torino; pp.314-315.319.491-502
- [5] <https://medicinaonline.co/2016/10/08/come-fatto-il-cervello/> consultato il 13/08/18
- [6] <https://www.docsity.com/it/2-anatomia-parte-2/618779/> consultato il 8/08/18
- [7] Ahlfors, S. P., Han, J., Belliveau, J. W., & Hämäläinen, M. S. (2010). “Sensitivity of MEG and EEG to source orientation”. *Brain Topography*, 23, 227–232.
- [8] Dugdale, D. E. (1993). “Essentials of electromagnetism”. New York, NY: American Institute of Physics
- [9] Jackson A.F., Bolger D.J. (2014) “The neurophysiological bases of EEG and EEG measurement: a review for the rest of us”. *Psychophysiology*, 51, 1061–1071
- [10] Ward L. M., (2003). “Synchronous neural oscillations and cognitive processes”. *TRENDS in Cognitive Sciences*, 7(12), 553.555.557.
- [11] <https://www.elettrocardiogramma.info/elettroencefalogramma/> consultato il 13/08/18
- [12] Klimesch, W. (1999). “EEG alpha and theta oscillations reflect cognitive and memory performance: a review and analysis”. *Brain Res. Rev.* 29, 169–195
- [13] Jensen, O. et al. (2002). “Oscillations in the alpha band (9–12 Hz) increase with memory load during retention in a short-term memory task”. *Cereb. Cortex* 12, 877–882
- [14] Herrmann C. S., et al. (2016). “EEG oscillations: From correlation to causality”. *International Journal of Psychophysiology* 103, 12–21
- [15] Lamberti C., (2017). “Elettrocardiografia”. *Dispense di Ingegneria Clinica*. Cap 2, pp.9-12
- [16] <https://clinicalgate.com/ecg-basics-waves-intervals-and-segments/> consultato il 08/08/18
- [17] Opthof, T (2000). “The normal range and determinants of the intrinsic heart rate in man”. *Cardiovasc.Res.* 45, 177–184.doi: 10.1016/S0008-6363(99)00322-3
- [18] Tortora, G. J., Derrickson, B. H. (2014). “Principles of Anatomy and Physiology. Hoboken,NJ:JohnWiley&Sons,Inc.
- [19] Shaffer F, et al. (2014). “A healthy heart is not a metronome: an integrative review of the heart’s anatomy and heart rate variability”. *Frontiers in Psychology*. doi: 10.3389/fpsyg.2014.01040, 1.2.5-10.15
- [20] http://www.rivistadipsichiatria.it/articoli.php?archivio=yes&vol_id=1069&id=11718 consultato il 13/08/18

- [21] Berntson, G.G., et al. (2008). “Cardiac autonomic balance versus regulatory capacity”. *Psychophysiology* 45, 643–652. doi: 10.1111/j.1469-8986.2008.00652
- [22] Eckberg, D.L., Eckberg, M.J. (1982). “Human sinus node responses to repetitive ramped carotid baroreceptor stimuli”. *Am.J.Physiol.* 242, H638–H644.
- [23] Malliani, A. (1995). “Association of heart rate variability components with physiological regulatory mechanisms,” in *HeartRateVariability*, Armonk, NY: Futura Publishing Company, Inc., 173–188.
- [24] Malińska M, Zużewicz K, Bugajska J Grabowski A (2015). “Heart rate variability (HRV) during virtual reality immersion” *International Journal of Occupational Safety and Ergonomics*: 21(1), pp.47.54
- [25] https://www.researchgate.net/figure/An-example-of-a-heart-rate-variability-power-spectrum-obtained-using-the-fast-Fourier_fig2_232599468 consultato il 13/08/18
- [26] <http://www.ingegneriaindustriale.unibo.it/it/servizi-e-strutture/laboratori/laboratorio-di-realta-virtuale#--> consultato il 13/08/18
- [27] <https://shop.openbci.com/collections/frontpage/products/cyton-daisy-biosensing-boards-16-channel?variant=38959256526> consultato il 13/08/18
- [28] <http://www.dei.unibo.it/it/attivita-didattica/laboratori-e-tirocini/laboratori-didattici/lib> consultato il 13/08/18
- [30] <http://www.braintronics.nl/pages/Productdatabase/Equipment/11BB%20EEG-1166.htm> consultato il 13/08/18
- [31] https://www.researchgate.net/figure/10-20-Electrode-Placement-System-The-location-of-the-Neurosky-electrode-used-in-the_fig6_279398111 consultato il 13/08/18
- [32] <https://it.mathworks.com/discovery/matlab-gui.html> consultato il 14/08/18
- [33] https://www.researchgate.net/figure/Frequency-response-of-notch-filter-with-50Hz-central-frequency-and-5Hz-half-width_fig3_232650750 consultato il 20/08/18

