

Report Homework FCI I - Gruppo 2

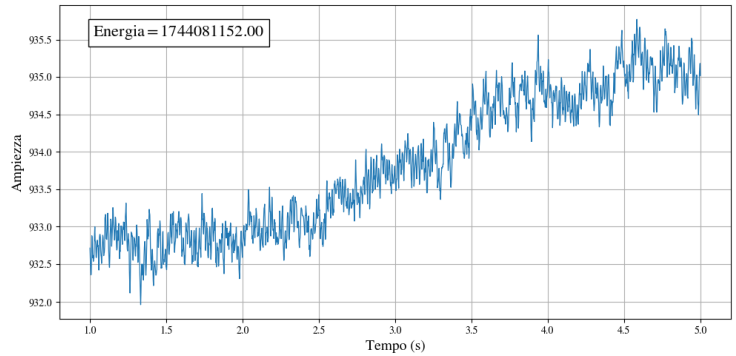
Davide Gaglione¹, Andrea Carbone, Alessandro Chitarrini e Matteo Crugliano

¹Rappresenta il gruppo in qualità di leader.

1 Definizione di segnali e grafici

Caricare il file trovato nella cartella “eeg-CP4-MI-LH-s01.mat” tramite la funzione `loadmat`. Il segnale è relativo al soggetto 1 mentre effettua il task di Motor Imagery della mano sinistra (Motor Imagery Left Hand). Si tratta del segnale registrato al sensore CP4. Considerare il segnale discreto soltanto nei campioni compresi tra $n_1 = 500$ e $n_2 = 2500$. Il passo temporale tra un campione e il successivo è di 2 ms. 1. Fare il grafico del segnale x_n ; 2. Calcolare l'energia di x_n e scriverla in un box all'interno del grafico precedente.

Per prima cosa, abbiamo importato il file `eeg-CP4-MI-LH-s01.mat` utilizzando la funzione `loadmat()`¹, che consente di caricare i dati da un file MATLAB. Il file contiene un segnale EEG registrato dal sensore CP4, relativo al soggetto 1 durante l'esecuzione del compito di Motor Imagery della mano sinistra. Dopo aver estratto il segnale, ci siamo concentrati unicamente sui campioni compresi tra $n_1 = 500$ e $n_2 = 2500$. Considerando un passo temporale tra i campioni pari a $t = 0.002s$ (2ms). In merito è stato possibile realizzare un asse temporale in secondi (raggiungendo un tempo totale analizzato di 4s), necessario per rappresentare correttamente il segnale in funzione del tempo. Successivamente, utilizzando la funzione `plot()`², abbiamo **rappresentato il segnale** x_n in funzione del tempo (visibile nella figura a destra). Infine, in questa prima sezione, abbiamo **calcolato l'energia** del segnale, definita come $E = \sum_{n=n_1}^{n_2} x_n^2$ e riportato il valore calcolato in un riquadro all'interno del grafico.



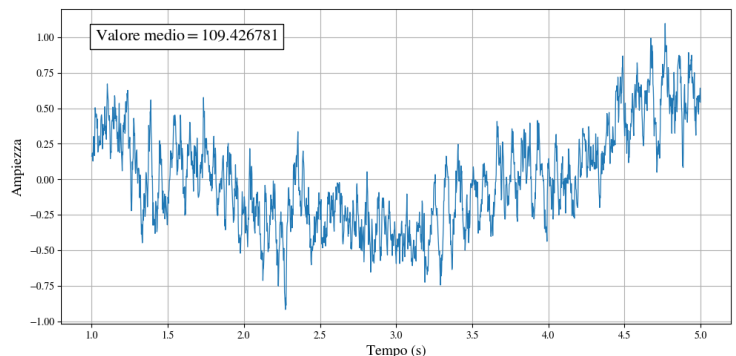
Segnale EEG del soggetto 1: Motor Imagery Left Hand (CP4)

2 Studio di una coppia di segnali

Caricare il file trovato nella cartella “eeg-C4-MI-LH-s01.mat”. Si tratta del segnale registrato dallo stesso soggetto per lo stesso task in corrispondenza del sensore C4. 3. Prendere il segnale soltanto nei campioni compresi tra $n_1 = 500$ e $n_2 = 2500$. Il passo temporale tra un campione e il successivo è di 2 ms. Calcolare il valore medio di questo segnale e sottrarlo ai valori del segnale. Rappresentare il grafico di questo segnale y_n (a cui è stato sottratto il suo valore medio); 4. Calcolare il coefficiente di correlazione tra il segnale originario x_n e y_n .

In questa seconda sezione abbiamo analizzato un altro segnale, questa volta registrato dal sensore C4 durante lo stesso esperimento per la mano sinistra. Il segnale è stato considerato nello stesso intervallo $[n_1, n_2]$ e passo (t) del primo sensore (CP4). Successivamente, abbiamo calcolato il valore medio del segnale e lo abbiamo sottratto da ciascun valore del segnale originale, ottenendo così il segnale grafato (y_n). Il passo successivo è stato il calcolo del coefficiente di correlazione tra il segnale x_n e il segnale y_n utilizzando la funzione `corrcoef()`³. Tale valore è pari a:

$$\rho_{x_n y_n} = 0.45150673558168236$$



Segnale EEG del soggetto 1: Motor Imagery Left Hand (C4)

¹della libreria `scipy.io`.

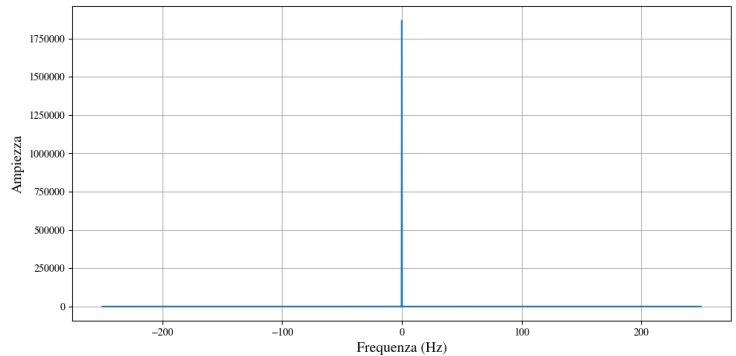
²della libreria `matplotlib.pyplot`. Abbiamo scelto questa funzione in alternativa a `stem()` per migliorare la leggibilità.

³della libreria `numpy`.

3 Studio del segnale in frequenza

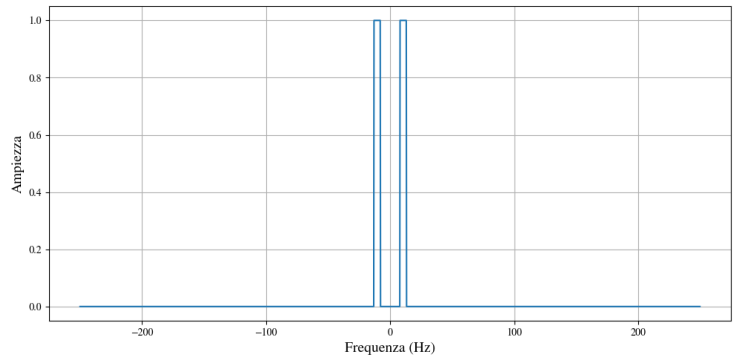
5. Calcola la trasformata di Fourier del segnale x_n usando la fast fourier transform (`fft`). Generare il vettore delle frequenze con `fftfreq`. Visualizzare il valore assoluto della trasformata di Fourier tramite `fftshift`. 6. Applicare un filtro passa banda nelle frequenze comprese in $\alpha = [8, 13]$ Hz. Rappresentare la risposta in frequenza del filtro e l'uscita del filtro nel tempo z_n .

Il segnale x_n è stato analizzato nel dominio della frequenza utilizzando la **trasformata di Fourier** discreta (`fft()`⁴). Dopo questo passaggio abbiamo generato il **vettore delle frequenze** con la funzione `fftfreq()`⁴, questo processo ha permesso di convertire il segnale dal dominio temporale a quello delle frequenze, consentendo di osservare come l'energia del segnale sia distribuita tra le diverse componenti frequenziali. Il modulo della trasformata è stato visualizzato dopo aver **centrato lo spettro** con l'operazione di `fftshift()`⁴, come visibile nel grafico a destra.



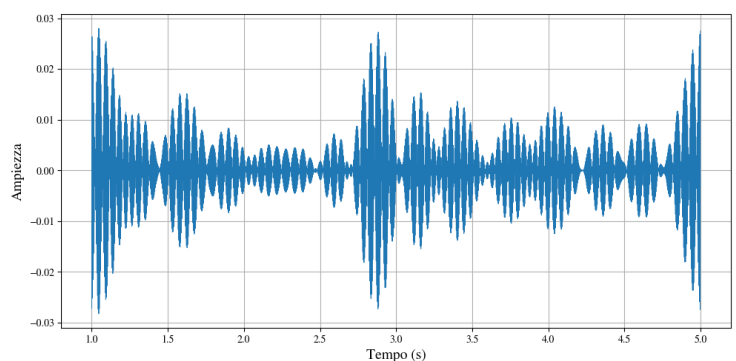
Valore assoluto della trasformata di Fourier

Per isolare una specifica banda di frequenze, abbiamo progettato un **filtro passa banda** ideale (`ideal_filter()`⁵), definito per lasciar passare solo le frequenze comprese tra 8 e 13 Hz. Questo filtro è stato implementato come una funzione a gradino che assume valore unitario nella banda desiderata e zero al di fuori. La sua **risposta in frequenza** $H(f)$ è stata graficata per confermare la selettività del filtro, come visibile in figura. Questo passaggio è cruciale per rimuovere componenti indesiderate dal segnale originale e preservare solo quelle di interesse. L'analisi del filtro è stata eseguita verificando il suo comportamento sullo spettro delle frequenze.



Risposta in frequenza del filtro passa banda

Successivamente, il filtro passa banda è stato applicato al segnale trasformato nel dominio della frequenza moltiplicando lo spettro $X(f)$ per la risposta del filtro $H(f)$. Questo passaggio consente di eliminare tutte le componenti frequenziali esterne alla banda compresa tra 8 e 13 Hz, preservando unicamente le frequenze desiderate. Il risultato, $Y_{BP}(f)$, rappresenta lo spettro filtrato. Attraverso la trasformata inversa di Fourier, calcolata con la funzione `ifft()`⁴, $Y_{BP}(f)$ è stato riportato nel dominio del tempo, ottenendo **l'uscita del filtro nel tempo** z_n , depurato dalle frequenze indesiderate, come visibile in figura.



Uscita del filtro nel tempo z_n

⁴della libreria `scipy.fft`.

```
5ideal_filter(sig, freqsig, f_low=None, f_high=None):
    H = np.ones_like(sig)
    if f_low is not None:
        H[np.abs(freqsig) < f_low] = 0
    if f_high is not None:
        H[np.abs(freqsig) > f_high] = 0
    return H
```

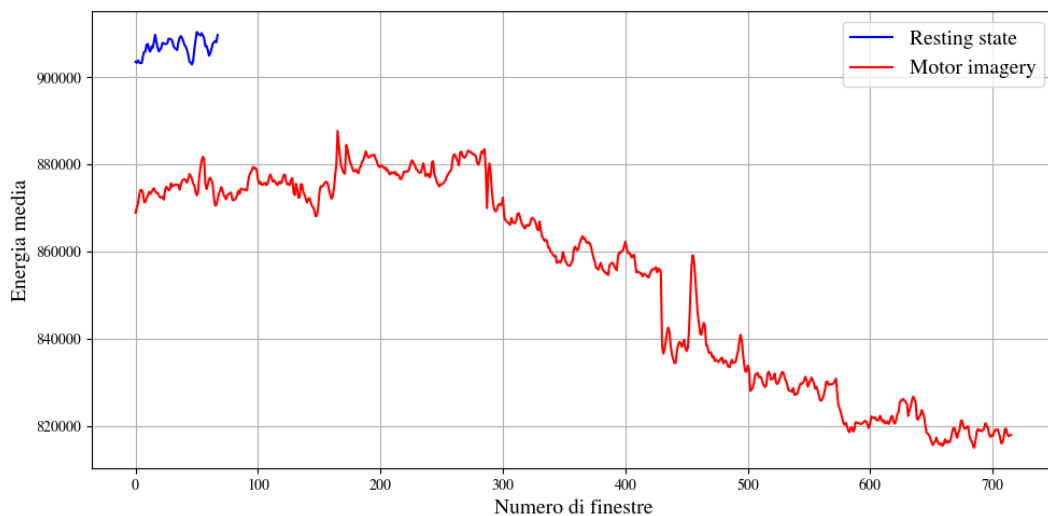
4 Domanda extra

Prendere tutti i campioni del segnale originario in “eeg-CP4_ML_LH_s01.mat” e dividerli in finestre di $N_f = 500$ campioni. Calcolare l’energia media per campione sulle diverse finestre e fare il grafico dell’energia (asse y) al variare delle finestre (asse x). Fare lo stesso con i dati di resting state file “eeg-CP4_rest_s01” (l’utente non sta facendo alcun task, è a riposo). Rappresentare le due curve sullo stesso grafico con due colori diversi per visualizzare come varia l’energia nel tempo nel caso in cui il soggetto si trovi in due condizioni cognitive diverse.

In questa analisi, abbiamo confrontato l’energia media per campione calcolata su finestre di 500 campioni dei segnali EEG registrati in due diverse condizioni cognitive: **motor imagery** (immaginazione motoria) e **resting state** (stato di riposo). Dopo aver caricato i segnali dai file forniti, ciascun segnale è stato suddiviso in finestre contigue di lunghezza $N_f = 500$. Per ogni finestra, è stata calcolata l’energia media come rapporto tra l’energia totale della finestra e il numero di campioni in essa contenuti, con la funzione:

```
def energia_media_f(signal, Nf=500):  
    val_en_med = []  
    num_finestra = len(signal) // Nf  
    for i in range(num_finestra):  
        finestra = signal[i * Nf:(i + 1) * Nf]  
        energia_media = energia(finestra) / Nf  
        val_en_med.append(energia_media)  
    return val_en_med
```

Il risultato per ciascun segnale è una serie di valori rappresentanti l’energia media delle finestre. Infine, abbiamo tracciato entrambe le serie sullo stesso grafico, utilizzando colori diversi per le due condizioni, evidenziando le differenze nell’andamento dell’energia in funzione delle finestre. Questo confronto permette di osservare come varia l’attività cerebrale tra una condizione attiva e una passiva, come si evince dall’illustrazione sottostante.



Confronto tra Resting State e Motor Imagery