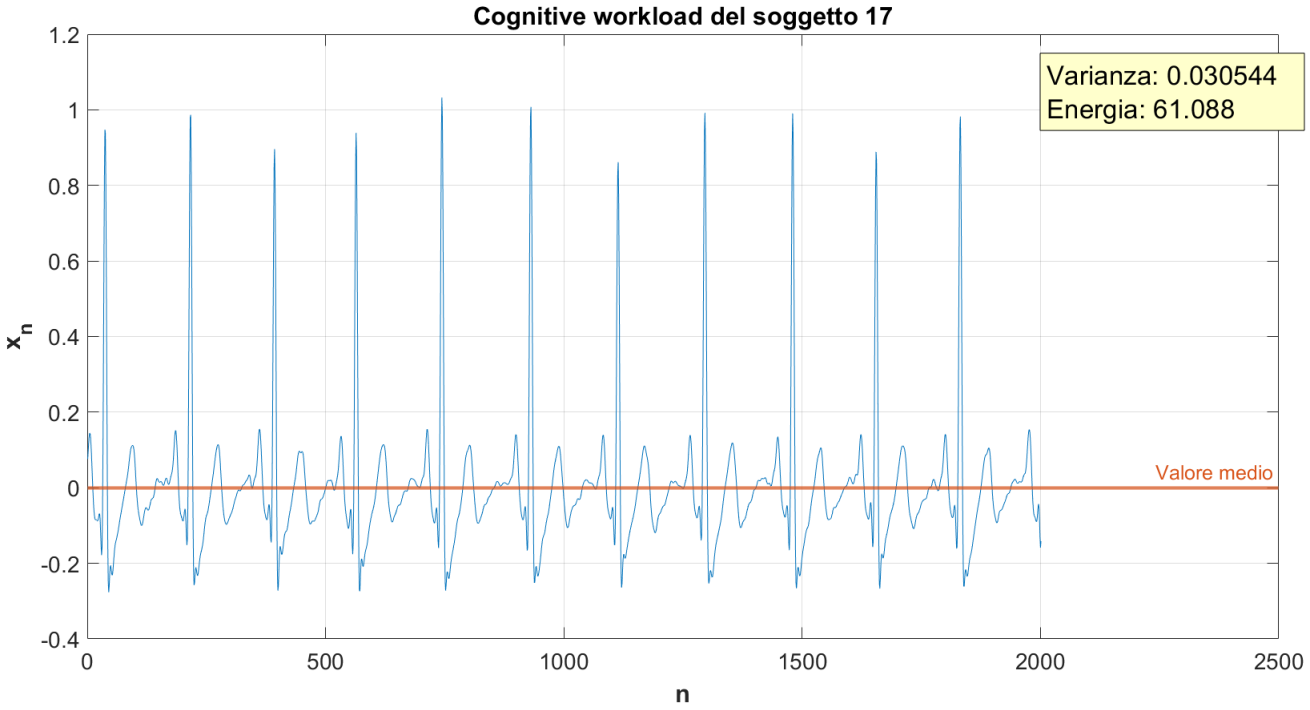
Gruppo: 9	Nome referente: Benedetta Fiorillo	Corso: Telecomunicazioni

Definizione del segnale e grafici

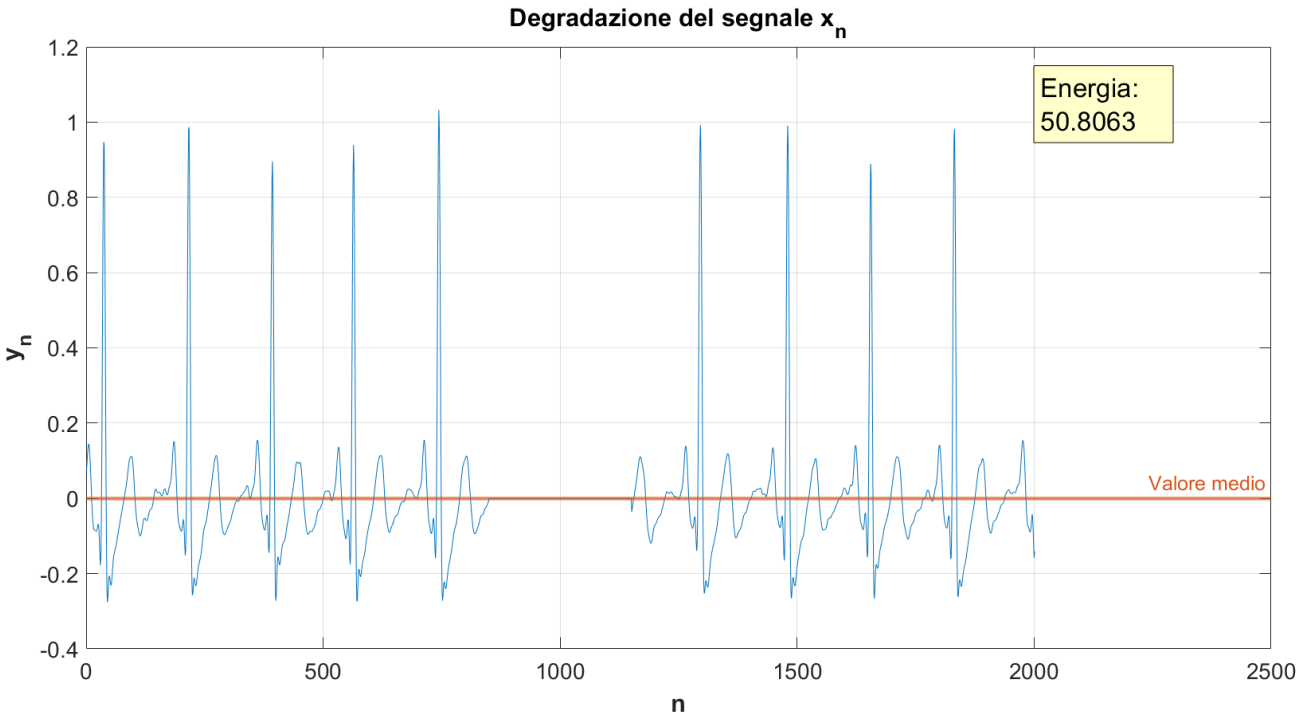
1) Il segnale di partenza deriva dall'elettrocardiogramma (ECG) misurato durante livelli più elevati di carico di lavoro cognitivo del paziente 17, filtrato e valutato opportunamente.
Sono stati estratti i campioni appartenenti all'intervallo [50000; 52000] con passo di campionamento 1 del segnale dato, con l'istruzione **`x_n = ECG(50000:52000, 1)`**. I valori sono stati assegnati ad un vettore colonna **`x_n`**, di dimensioni 2001x1. Il segnale **`x_n`** è stato poi graficato con la funzione **`plot(x_n)`** per maggiore resa grafica.

2-3) Per il calcolo del valore medio (pari a -2.0328×10^{-4}) e della varianza si è scelto di usare rispettivamente le funzioni **`mean(x_n)`** e **`var(x_n)`**.
Dato che il segnale è reale, si è omesso il modulo; quindi, il calcolo dell'energia è stato fatto con la funzione **`sum(x_n.^2)`**.



Degradazione del segnale e grafico

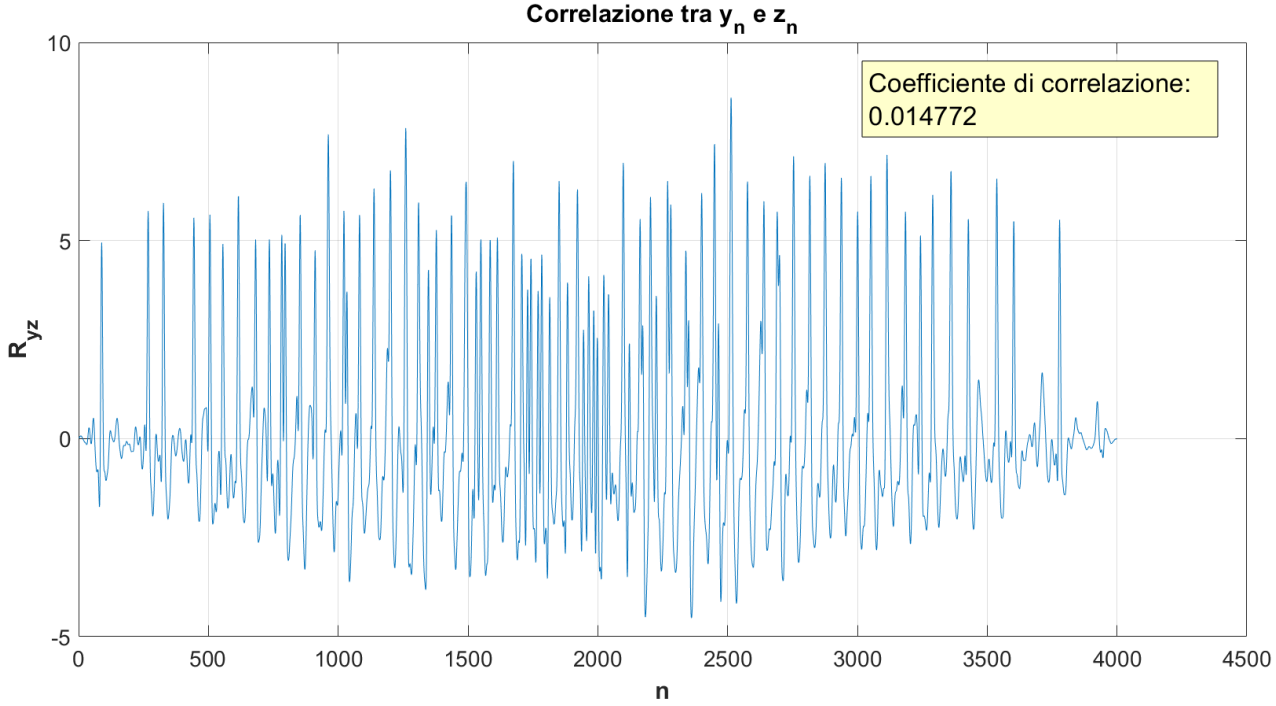
4) Dato **`x_n`**, abbiamo eliminato una percentuale di campioni pari al 15% (corrispondente a 300.15 campioni). Il numero di campioni da azzerare deve essere un numero intero, per questo motivo si è deciso di considerarne 301.
Per la degradazione si è scelto un intervallo "centrale" rispetto alla durata del segnale, ovvero [850; 1150].
Per la realizzazione si è deciso di usare un ciclo for sulla variabile i, in modo che, ad ogni iterazione, il corrispondente campione i-esimo fosse messo a zero.
Sono stati poi assegnati i valori al nuovo segnale **`y_n`** che è stato poi graficato con la funzione plot (come al punto 1). Anche in questo caso, il valore medio e l'energia sono stati calcolati con le stesse modalità del punto 1, ottenendo rispettivamente i valori -4.9219×10^{-4} e 50.8063.
Si osserva che l'azzeramento dei campioni ha causato una riduzione dell'energia del segnale **`x_n`**.



Calcolo correlazione e coefficiente di correlazione

5) Il segnale in esame questa volta è l'elettrocardiogramma (ECG) misurato durante livelli più elevati di carico di lavoro cognitivo del soggetto 1, nuovamente filtrato e trattato opportunamente. Il segnale **`z_n`** è stato poi ricavato nello stesso modo del segnale **`x_n`**. Per calcolare la correlazione tra **`y_n`** e **`z_n`** si è fatto riferimento alla funzione **`xcorr(y_n, z_n)`**, che restituisce un vettore colonna di dimensioni 4001x1 – come ci si aspettava, la correlazione ha durata pari alla somma delle durate dei due segnali. Il segnale risultante è stato poi graficato, come nei due punti precedenti, con la funzione **`plot`**.

6) Per calcolare il coefficiente di correlazione si è fatto uso della funzione **`corr2(y_n, z_n)`**. Essendo il valore di quest'ultimo prossimo allo 0, possiamo dedurre che i due segnali hanno una scarsa similarità.



Impostazioni di layout delle figure

Le figure riportate sono state inserite in una finestra suddivisa in tre sottotrame che permettono la visione complessiva e completa di quanto fatto.

Di seguito sono riportate le funzioni grafiche per aumentare la leggibilità delle figure:

- **`title`** per aggiungere il titolo ad ogni sottotrama
- **`xlabel`** e **`ylabel`** per aggiungere il nome degli assi e impostare il font
- **`grid on`** per mettere sullo sfondo una griglia
- **`hold on`** per mantenere il contenuto attuale della finestra grafica in modo tale che i grafici successivi si sovrappongano a quelli già presenti
- **`get(0, 'ScreenSize');`**
  `larghezza_schermo = schermo(3);`
  `altezza_schermo = schermo(4);`
  `set(gcf, 'Position', [1, 1, larghezza_schermo, altezza_schermo]);`
  per far apparire la figura a schermo intero
- **`set(gca,'FontSize',17)`** per la generazione delle immagini al fine di aumentare il carattere dei valori sugli assi

Domanda Extra

Di seguito si è deciso di riportare due versioni, una con la funzione offerta **`findpeaks()`** ed una in cui quest'ultima funzione viene implementata manualmente da noi. In entrambe le versioni la soglia scelta è 0.7 ed è stata calcolata la distanza tra il quarto e il quinto picco, ottenendo un discostamento di 0.5 tra le due soluzioni (i rispettivi risultati sono riportati di seguito).

- Versione con findpeaks():

```
[peak_value, peak_location] = findpeaks(x_n, 'MinPeakHeight', 0.7);
singola_distanza = peak_location(5) - peak_location(4);
fprintf("Distanza tra quarto e quinto picco:" ); disp(singola_distanza);
% risultato = 180
```

- Versione "manuale":

In "array_pos_picchi" sono riportate le posizioni dei picchi; ad ogni picco corrisponde la sua posizione media.

```
soglia=0.7; c=1; n_entry_arraypicchi=0; array_pos_picchi=zeros(10000);
while (c < length(x_n))
    while (x_n(c)<soglia && c<length(x_n))
        c=c+1;
    end
    if(c==length(x_n))
        break;
    else
        n_entry_arraypicchi=n_entry_arraypicchi+1;
        sum_pos_picchi=0;
        nover=0;
    end
    while (x_n(c)>=soglia && c<length(x_n))
        nover=nover+1;
        sum_pos_picchi=sum_pos_picchi+c;
        c=c+1;
    end
    array_pos_picchi(n_entry_arraypicchi)=(sum_pos_picchi/nover);
end
distanza=array_pos_picchi(5)-array_pos_picchi(4);
fprintf("La distanza tra i picchi 4 e 5 è: %f\n",distanza);
% risultato = 179.5
```