Introduzione a Matlab

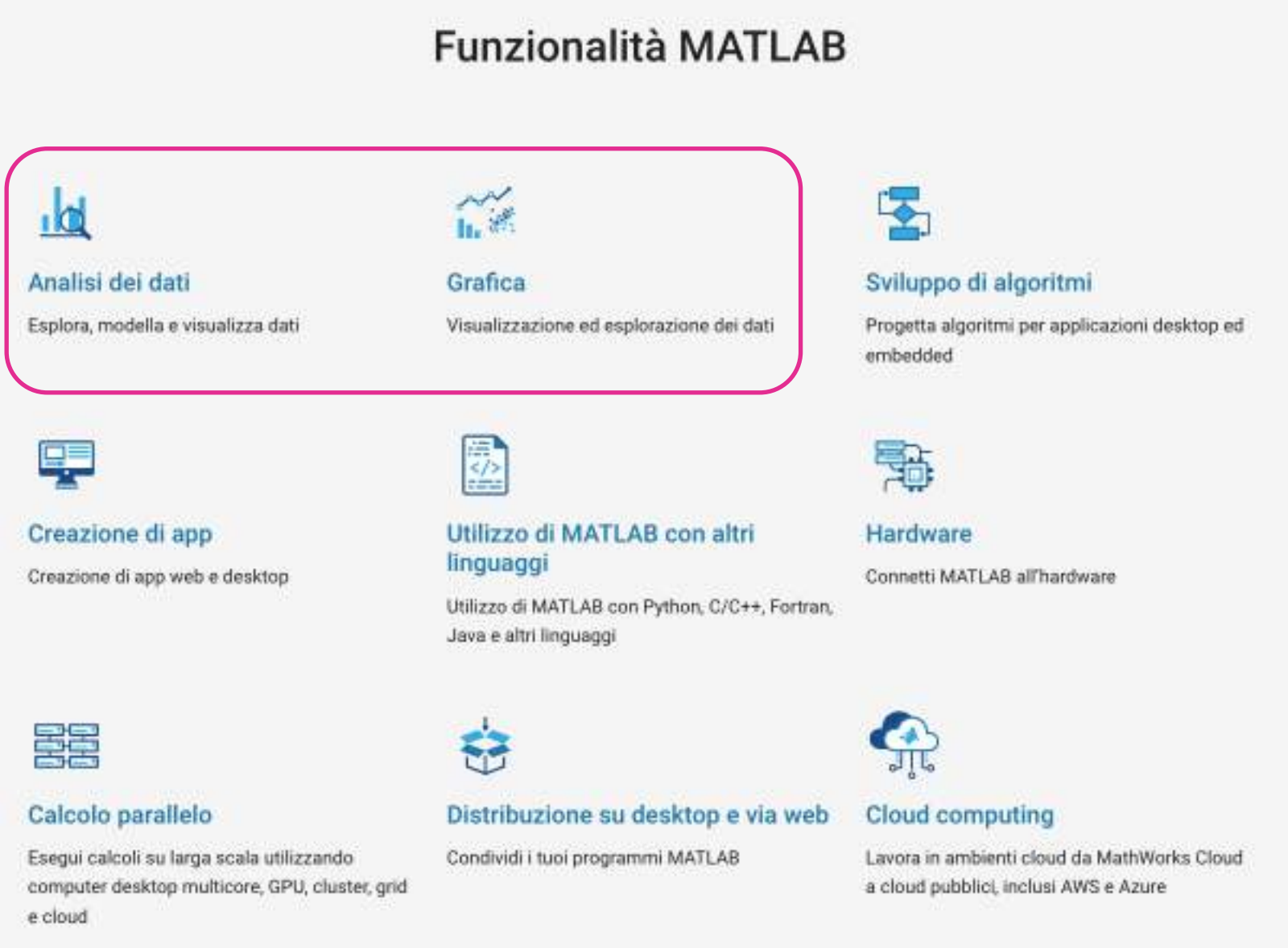
Tutor: Dr. Giovanna Nordio e Giulia Vallini

Prof. Mattia Veronese
Email: mattia.veronese@unipd.it
Dipartimento di Ingegneria dell'Informazione

Ricevimento: su appuntamento (e-mail)
Edificio DEI/A, piano 2o, stanza 214

A cosa serve MATLAB?

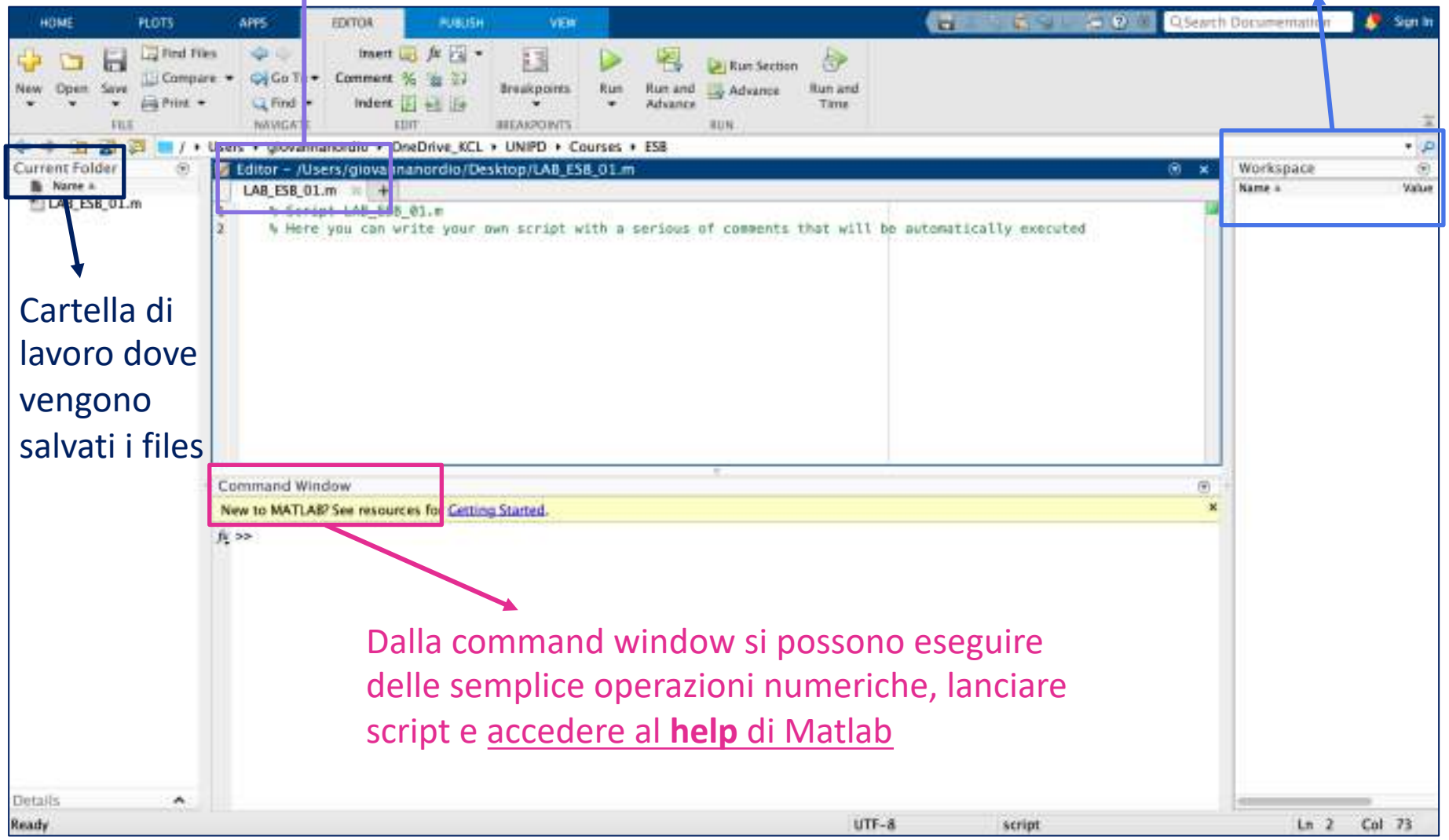
MATLAB e' un linguaggio di calcolo matriciale e vettoriale



L'interfaccia grafica di Matlab

Spazio di lavoro dove vengono memorizzate le variabili create o importate in Matlab da file/programmi esterni

Matlab script



Cartella di lavoro dove vengono salvati i files

Dalla command window si possono eseguire delle semplice operazioni numeriche, lanciare script e accedere al **help** di Matlab

Matlab tutorials

Per chi non ha esperienza con Matlab:

https://matlabacademy.mathworks.com/

MATLAB Onramp
MATLAB Fundamentals

N.B. per accedere dovete creare un account Mathworks (con la vostra mail unipd)

Altri tutorials utili:

- https://people.ucalgary.ca/~ranga/enel563/matlab_tutorial.pdf (STESSO DEL LIBRO Biomedical Signal Analysis)
- https://pselab.chem.polimi.it/wp-content/uploads/2017/03/SECDIC-Tutorial-Matlab2.pdf

Script e funzione

SCRIPT

Uno script è un file .m che contiene piú linee di comando e operazioni, eseguibili automaticamente tramite chiamata diretta.

FUNZIONE

Una funzione è un programma che, avendo a disposizione una serie di dati di input, fornisce una serie di dati di output, effettuando delle operazioni. Una funzione esterna permette di salvare una serie di operazioni che vengono ripetute frequentemente, richiamandola negli script SOLO quando necessario (es. EoS, sistemi di equazioni algebriche o differenziali,...).

```
function [OUTPUT_1,…,OUTPUT_N] = myfunction(INPUT_1,…,INPUT_M)
```

Alcuni comandi utili

Consiglio - Iniziare uno script con:

clear all: pulisce il Workspace, cancellando tutte le variabili salvate
close all: chiude i grafici aperti
clc: pulisce i comandi scritti nella Command Window

Alcune strategie per scrivere uno script/funzione ordinata ed efficiente:

- Commenta il piu' possibile il tuo codice
- Raggruppa linee di codice in base alla loro funzionalita'
- Fai il backup del tuo codice regolarmente
- Usa nomi di variabili che hanno senso

NB Matlab dispone di una ricca libreria di funzioni, a cui si può accedere tramite **help menu** (help *#nomefunzione*), oppure tramite documentazione online.

Calcolo matriciale e vettoriale

MATLAB è un linguaggio di calcolo matriciale e vettoriale

Queste sono solo alcune delle operazioni utili da ricordare:

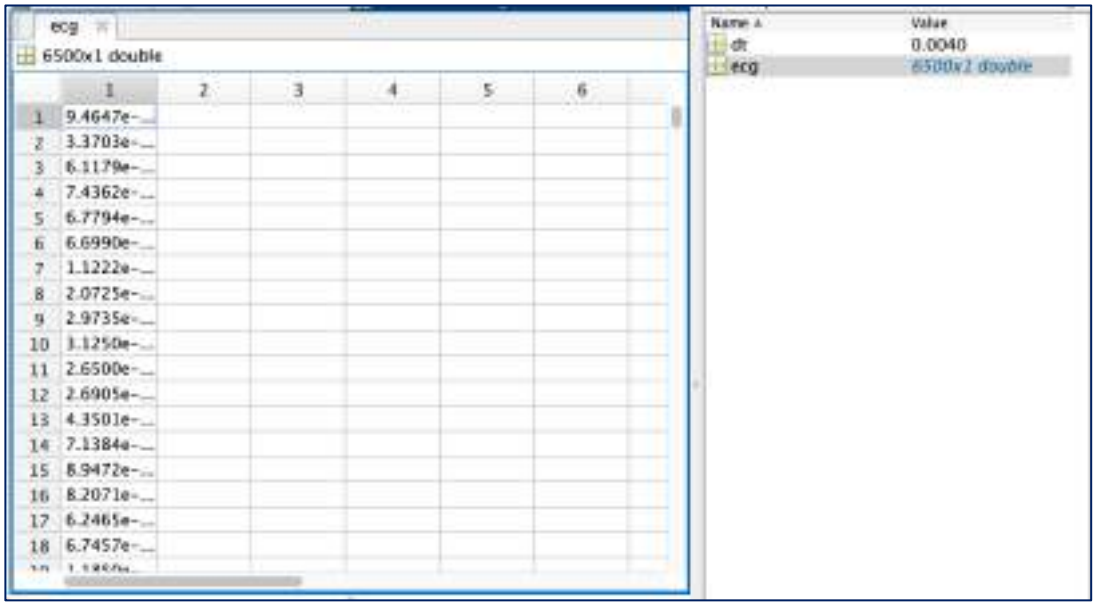
OPERATION	MATLAB COMMAND
Matrice 2x3	>> A = [1 5 7; 2 9 8] 1 5 7 2 9 8
Matrice vuota 2x3	>> A = zeros(2,3) 0 0 0 0 0 0
Vettore colonna vuoto 3x1	>> v = zeros(3,1) 0 0 0
Vettore riga di solo uno 1x3	>> v = ones(1,3) 1 1 1
Vettore di punti equispaziati	>> v = 1:5 1 2 3 4 5 >> v = -1.1:0.4:0.2 -1.1000 -0.7000 -0.3000 0.1000

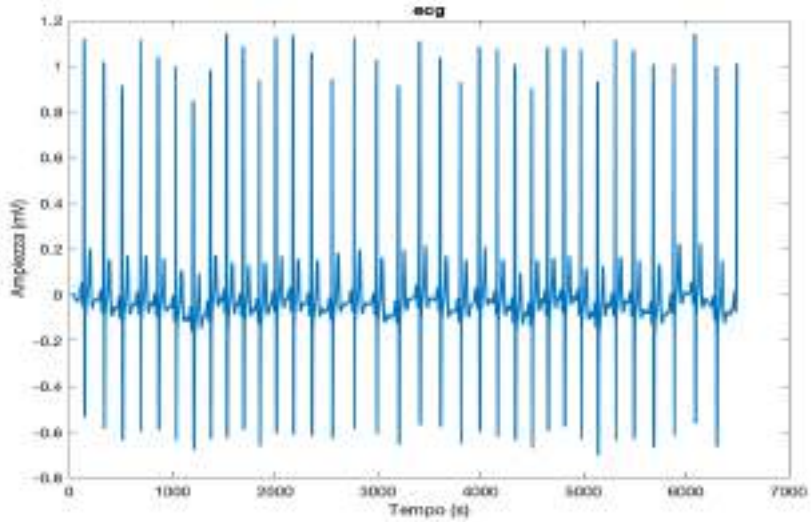
Calcolo matriciale e vettoriale

Un segnale biomedico campionato è un classico esempio di array (vettore).

Caso ECG

ECG misurato è un vettore lungo 6500 campioni





Rappresentazione grafica di un segnale

Matlab offre una ricca libreria per rappresentare graficamente un segnale.
Per un segnale bidimensionale, in analogia ad un segnale di una variabile dove l'ordinata è un vettore, che rappresenta la funzione valutata per N punti dell'ascissa, l'ordinata è una matrice, che rappresenta la funzione valutata per NxM punti delle due variabili indipendenti x e y.

```
plot(x, y, 'objects', value)
```

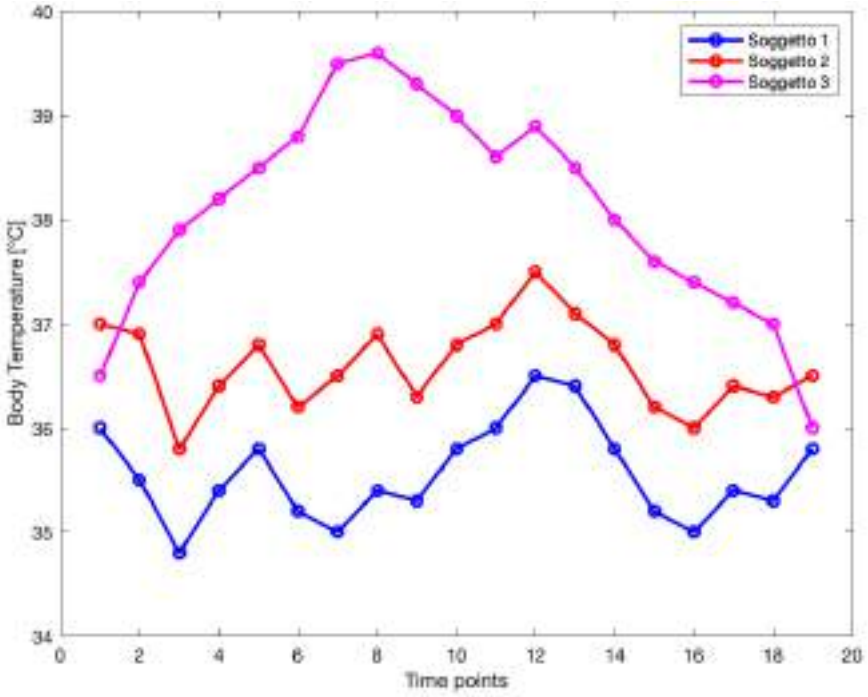
Proprietà:

- linewidth
- markersize
- markeredgecolor
- markerfacecolor

Valore: dipende dalla proprietà

- Dimensione in punti
- Colori

```
% Plot
figure(1)
plot(thermo(:,1), '-bo', 'LineWidth',2)
hold on
plot(thermo(:,2), '-ro', 'LineWidth',2)
hold on
plot(thermo(:,3), '-mo', 'LineWidth',2)
xlabel('Time points')
ylabel('Body Temperature [°C]')
legend('Soggetto 1','Soggetto 2', 'Soggetto 3')
```

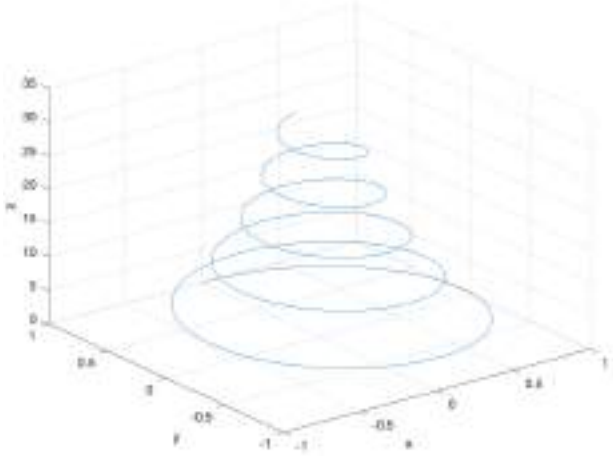


Rappresentazione grafica di un segnale

Per segnali tridimensionali:

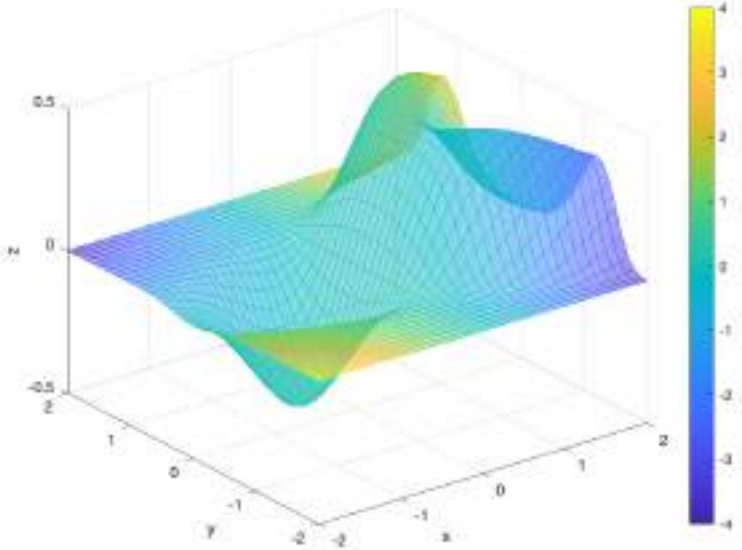
- `plot3` → per tracciare una linea in uno spazio tridimensionale

```
figure(2)
t = [0:pi/50:10*pi];
plot3(exp(-0.05*t).*sin(t),exp(-0.05*t).*cos(t),t)
xlabel('x')
ylabel('y')
zlabel('z')
grid
```



- `mesh` → per tracciare una superficie 3D

```
figure(3)
[X,Y] = meshgrid(-2:0.1:2);
Z = X.*exp(-((X-Y.^2).^2+Y.^2));
C = X.*Y;
s = mesh(X,Y,Z,C,'FaceAlpha','0.5')
s.FaceColor = 'flat';
xlabel('x')
ylabel('y')
zlabel('z')
colorbar
```



Esercizio 1

Creare il vettore riga v = (5, 9, -6, 0, 0.2, 3.6) e il vettore colonna w = (0, 0.1, -5, -3, 2.3, 9)

1. Salvare in A la somma di v e w
2. Salvare in B il prodotto scalare di v per w (usare il commando **dot**)
3. Salvare in C gli elementi di posto pari di v
4. Salvare in D cinque copie del vettore v (suggerimento: usare il comando **kron**)
5. Salvare in E gli elementi di posto multiplo di 4 di D
6. Salvare in F la matrice w*v
7. Salvare in G la matrice costituita dalla seconda fino alla quarta riga di F e dalla prima fino alla terza Colonna di F

Soluzioni - Esercizio 1

```
% Vettore riga v
v = [5, 9, -6, 0, 0.2, 3.6]; % Altrimenti v = [5 9 -6 0 -0.2 3.6];

% Vettore colonna W
w = [0, 0.1, -5, -3, 2.3, 9]'; % Altrimenti v = [0; 0.1; -5; -3; 2.3; 9];

% Salvare in A la somma del vettore riga v e  del vettore colonna w
A = v' + w;

% Salvare in B il prodotto scalare di v per w (usare il commando dot)
B = dot(v',w);

%Salvare in C gli elementi di posto pari di v
C = v(2:2:end);

%Salvare in D cinque copie del vettore v
D = kron(ones(5,1),v);

%Salvare in E gli elementi di posto multiplo di 4 di D
E = D(4:4:end);

%Salvare in F la matrice w*v
F = w*v;

%Salvare in G la matrice costituita dalla seconda fino alla quarta riga di
%F e dalla prima fino alla terza Colonna di F
G = F(2:4,1:3);
```

Esercizio 2

Il file DATA_Lab03_Es02.xlsx contiene una tabella di 3 colonne e 130 righe, dove la prima colonna corrisponde alla temperatura corporea (in Fahrenheit) di 130 soggetti, la seconda colonna è il sesso dei soggetti (0=uomo, 1=donna), e la terza colonna e' il battito cardiaco dei 130 soggetti.

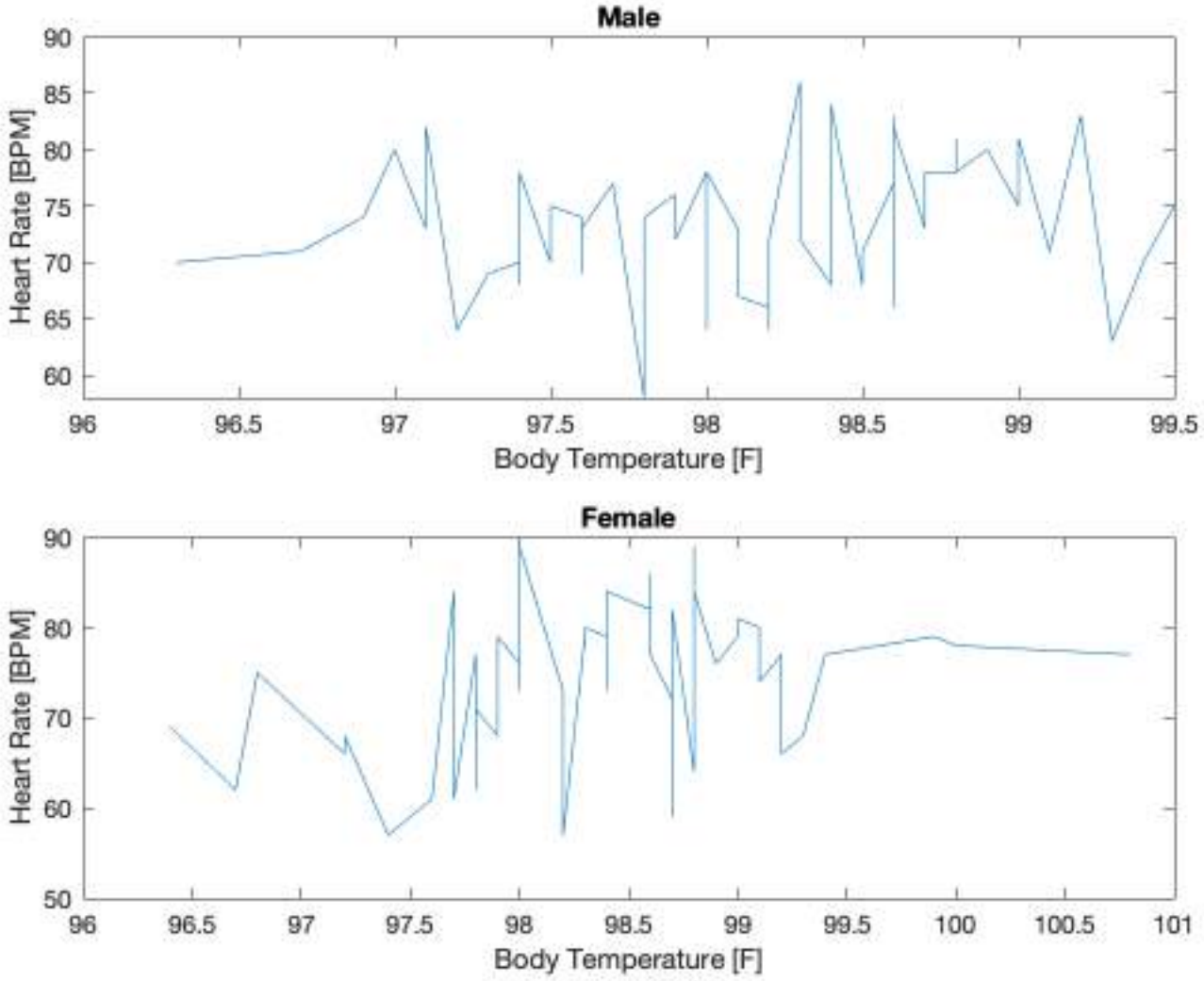
1. Carica il file contenente i dati

   N.B. Per importare un file in Matlab è possibile usare una di queste alternative

   - **load filename.extension**
   - **xlsread('filename')** per file excel
   - **importdata('filename')**

2. Crea due matrici (130x2), una per gli uomini e una per le donne, con la corrispondente temperatura e battito cardiaco (suggerimento: usa il comando **find**)
3. Rappresenta graficamente la temperatura corporea in funzione del battito cardiaco per gli uomini e per le donne in due grafici separati (usa il comando **subplot**)

Soluzioni – Esercizio 2



Esercizio 3

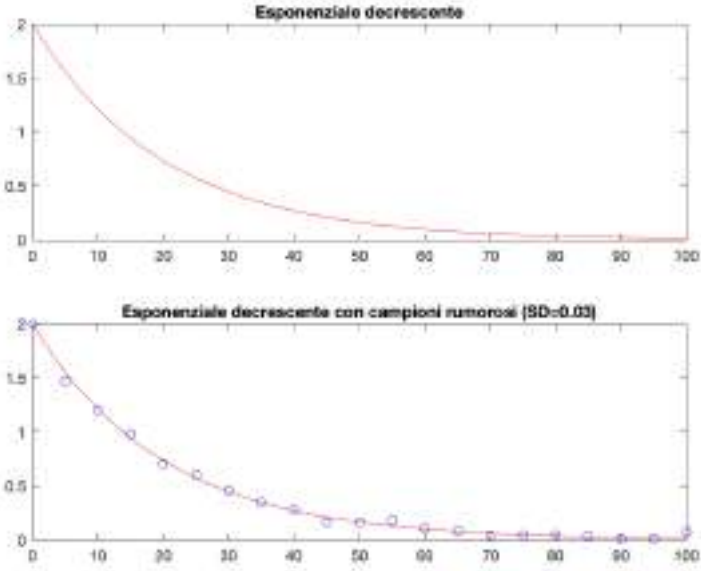
1. Scrivere un M-file che disegni con una linea continua rossa la funzione **$f(t)=2e^{(-0.05t)}$** nell'intervallo [0,100] (rappresentare f(t) con 500 punti equispaziati; usare il comando **linspace**)

2. Rappresentare f(t) nel riquadro superiore di una figura a due riquadri (suggerimento: usare **subplot**). Nel riquadro inferiore rappresentare di nuovo la stessa funzione con sovrapposti, rappresentati da cerchietti blu, dei suoi campioni rumorosi generati artificialmente ai tempi (0, 5, 10, 15, …100). Per generare i campioni considerare rumore gaussiano additivo a media nulla e standard deviation 0.03 (suggerimento: usare funzione **randn**). Intitolare i grafici rispettivamente con *'Esponenziale decrescente'* ed *'Esponenziale decrescente con campioni rumorosi (SD=0.03)'.*

3. Acquisire da tastiera (suggerimento: usare l'istruzione input) un valore positivo (suggerimento: usare una struttura **while … end** per controllare che l'utente immetta un valore SD strettamente positivo) da usare come standard deviation del rumore additivo da usare nella generazione dei campioni rumorosi di f(t). Intitolare i grafici dei due riquadri rispettivamente con "Esponenziale decrescente" e "Esponenziale decrescente con campioni rumorosi (SD=__)" (far apparire nel titolo del secondo riquadro il valore della SD immesso da tastiera, suggerimento: usare **num2str**)

Soluzioni - Esercizio 3

```
t = linspace(0,100,500)';
f = 2*exp(-0.05*t);
subplot(2,1,1)
plot(t, f, 'r')
title('Esponenziale decrescente')

ts = [0:5:100]';
fs = 2*exp(-0.05*ts);
vs = 0.03*randn(length(fs),1);
ys = fs + vs;

subplot(2,1,2)
plot(t, f, 'r')
hold on
plot(ts, ys, 'ob')
title('Esponenziale decrescente con campioni rumorosi (SD=0.03)')
```




```
sd = input('Inserisci un valore di SD: ');
while sd <= 0

    sd = input('Dammi un valore positivo di SD: ');

end % while

figure(2)
subplot(2,1,1)
plot(t, f, 'r')
title('Esponenziale decrescente')

ts = [0:5:100]';
fs = 2*exp(-0.05*ts);
vs = sd*randn(length(fs),1);
ys = fs + vs;

subplot(2,1,2)
plot(t, f, 'r',ts, ys, 'ob')
title(['Esponenziale decrescente con campioni rumorosi (SD='+convertCharsToStrings(num2str(sd))+')'])
```

Esercizio 4 – da fare a casa

Data una matrice A, le cui dimensioni sono *m* per le righe ed *n* per le colonne, creare una matrice B contenente gli stessi elementi della matrice A, ma con le righe pari ordinate in modo crescente e le righe dispari ordinate in maniera decrescente.

Suggerimenti:

- crea una funzione ordina_crescente
- crea una funzione ordina_decrescente

Soluzioni - Esercizio 4

```
function [ M ] = ordina_crescente( M, r, c )
%ordina_crescente funzione che implementa l'algoritmo del selection sort
%   La funzione accetta 3 parametri (la matrice, l'indice della riga da
%   ordinare e il numero di colonne) e restituisce la matrice con la
%   r-esima riga ordinata in modo crescente

    for i = 1 : c-1
        i_min = i;
        j = i+1;
        for j = j : c
            if M(r, j) < M(r, i_min)
                % Se vero, aggiorno l'indice dell'elemento minimo
                i_min = j;
            end
        end

        % Blocco di codice relativo allo scambio
        temp = M(r, i);
        M(r, i) = M(r, i_min);
        M(r, i_min) = temp;
    end
end
```

```
function [ M ] = ordina_decrescente( M, r, c )
 %ordina_decrescente funzione che implementa l'algoritmo del selection sort
%   La funzione accetta 3 parametri (la matrice, l'indice della riga da
%   ordinare e il numero di colonne) e restituisce la matrice con la
%   r-esima riga ordinata in modo decrescente

    for i = 1 : c-1
        i_max = i;
        j = i+1;
        for j = j : c
            if M(r, j) > M(r, i_max)
                i_max = j;
            end
        end

        temp = M(r, i);
        M(r, i) = M(r, i_max);
        M(r, i_max) = temp;
    end
end
```

Soluzioni - Esercizio 4

```
%% Esercizio 4
% Input la dimensione righe della matrice A
flag=true;
while flag==true
    m = input('Inserisci numero di righe per la matrice A:');
    flag = m<=0;
end

% Input la dimensione colonne della matrice A
flag=true;
while flag==true
    n = input('Inserisci numero di colonne per la matrice A:');
    flag = n<=0;
end

% Input gli elementi della matrice A
disp('Inserisci i valori della matrice A (scrivi un valore e premi invio)')
A=[];
for i=1:m
    for j=1:n
        A(i,j)=input('');
    end
end

% Creo la matrice B esattamente uguale alla matrice A
B = A;

for i = 1 : m
    if mod(i,2)==0
        % Caso riga pari
        B = ordina_crescente(B, i, n);
    else
        % Caso riga dispari
        B = ordina_decrescente(B, i, n);
    end
end

disp(A)
disp(B);
```

Rappresentazioni di segnali biomedici

Tutor: Dr. Giovanna Nordio e Giulia Vallini

Prof. Mattia Veronese
Email: mattia.veronese@unipd.it
Dipartimento di Ingegneria dell'Informazione

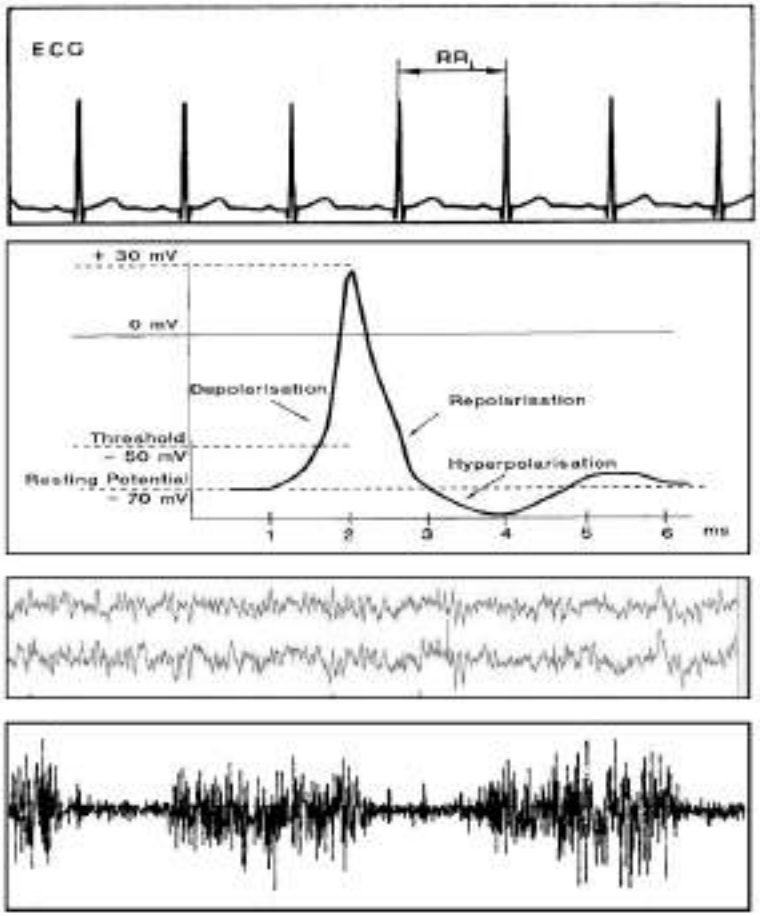
Ricevimento: su appuntamento (e-mail)
Edificio DEI/A, piano 2o, stanza 214

Segnali bioelettrici

I segnali bioelettrici rappresentano l'attività elettrica di un sistema biomedico.

La rappresentazione grafica e l'analisi di un segnale bioelettrico permettono di individuare eventuali anomalie che possono essere indice di un malfunzionamento del sistema biomedico corrispondente.

È quindi fondamentale capire come rappresentare e analizzare i segnali biomedici.



ECG

Potenziale d'azione

EEG

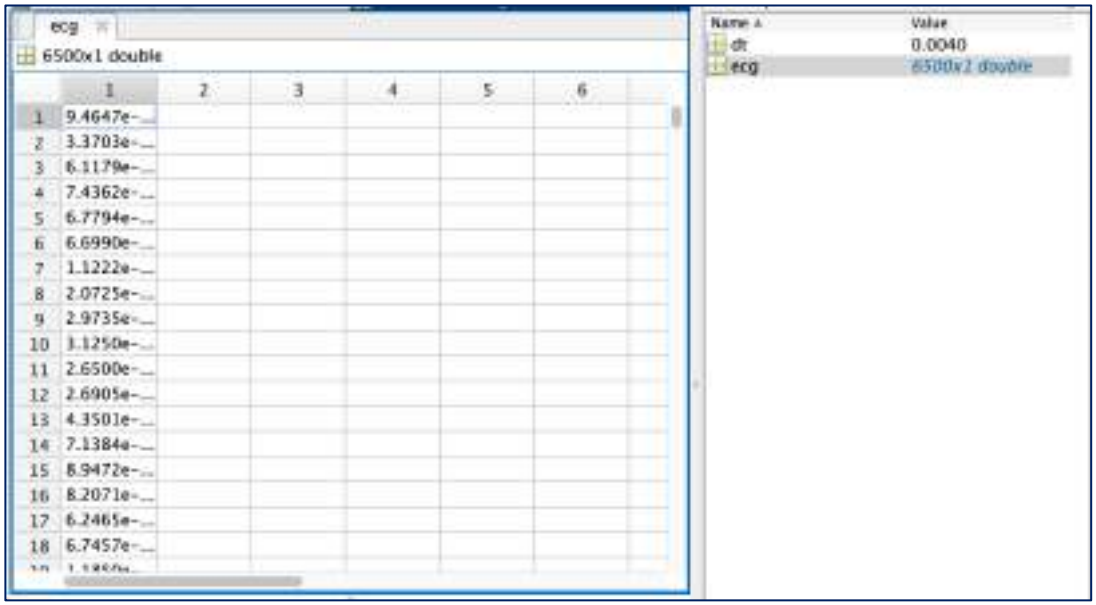
EMG

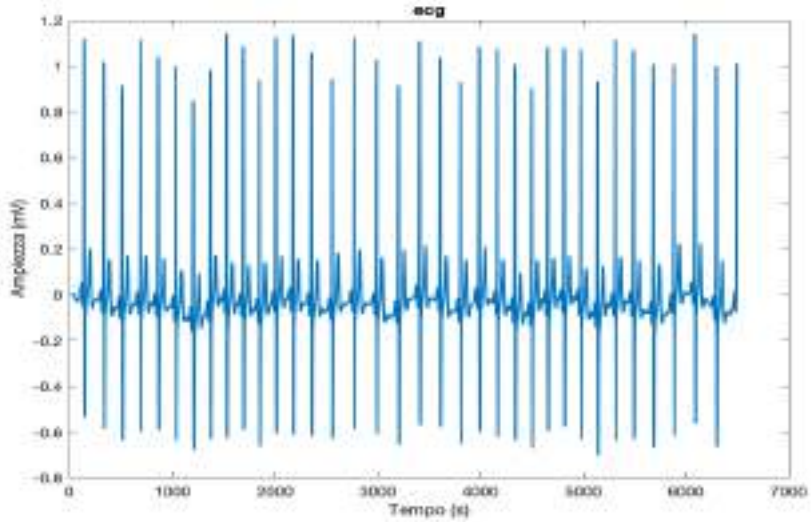
Segnali bioelettrici

Un segnale biomedico campionato è un classico esempio di array (vettore).

Caso ECG

ECG misurato è un vettore lungo 6500 campioni

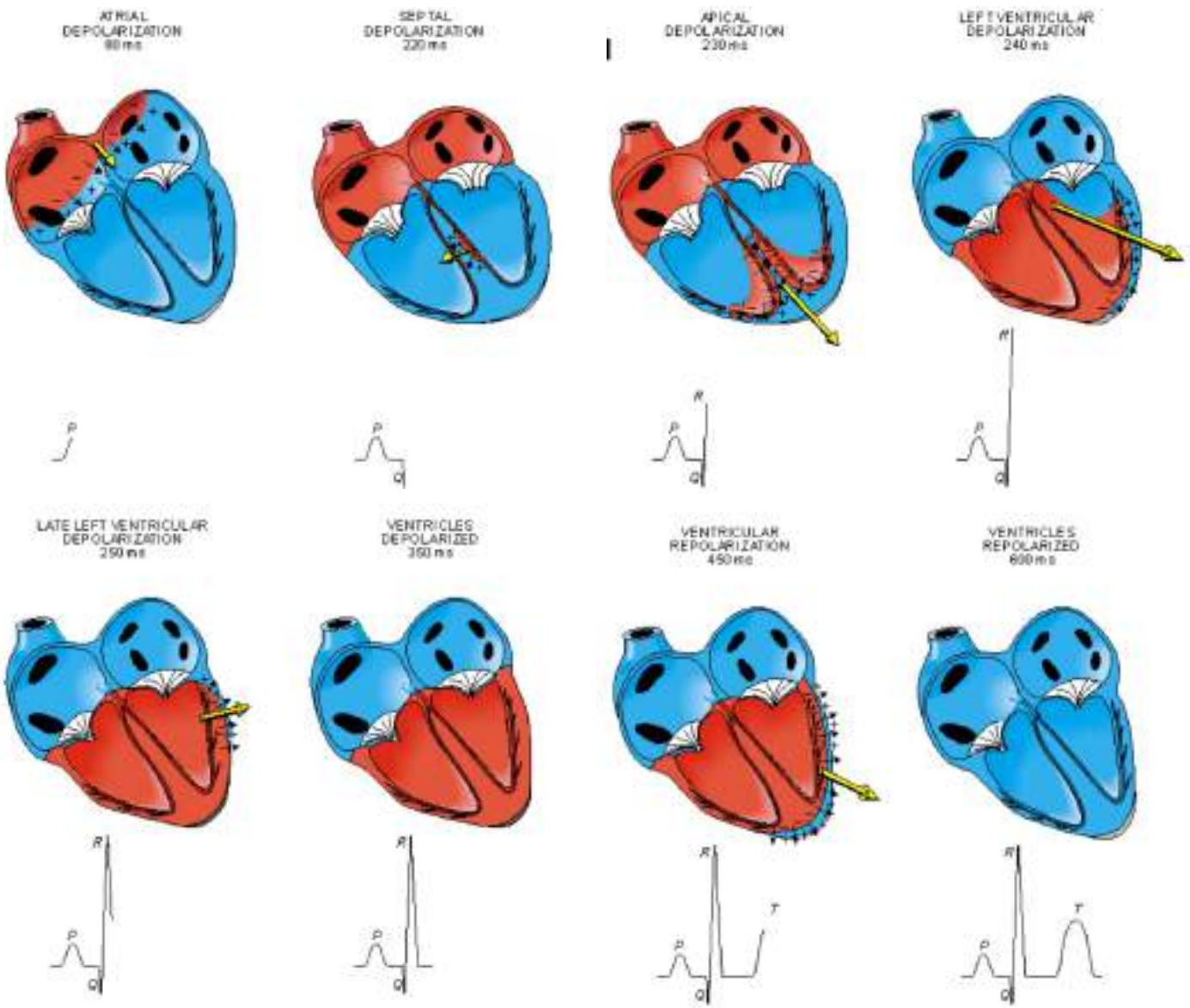




Obiettivo del laboratorio

L’obbiettivo di questo laboratorio è di utilizzare Matlab per rappresentare i segnali bioelettrici e capire come si estrapolano informazioni di interesse. In particolare, lavoreremo con i segnali bioelettrici ECG e EMG.

Il segnale ECG



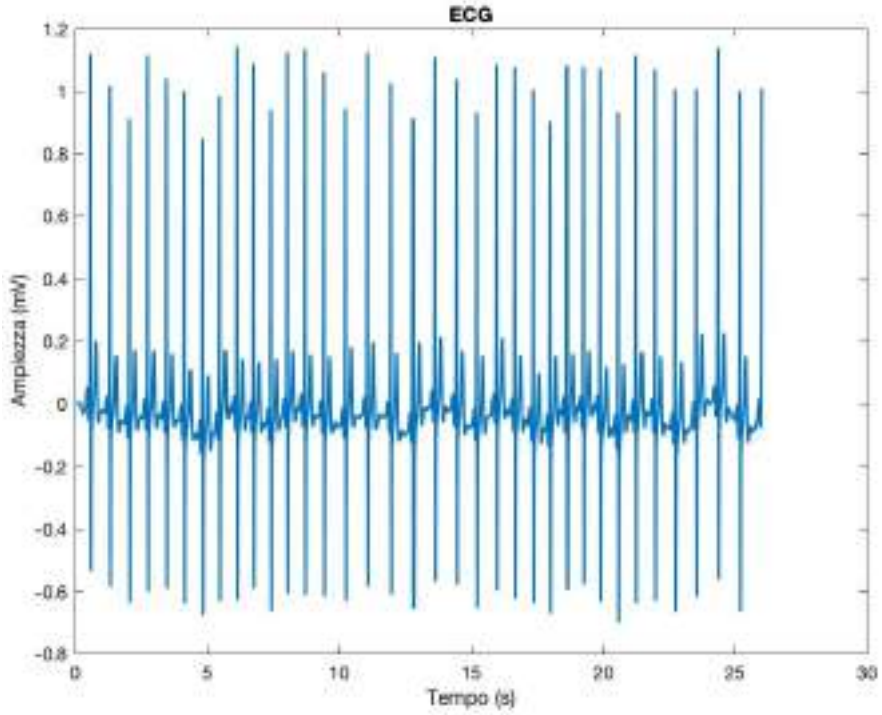
Esercizio 1

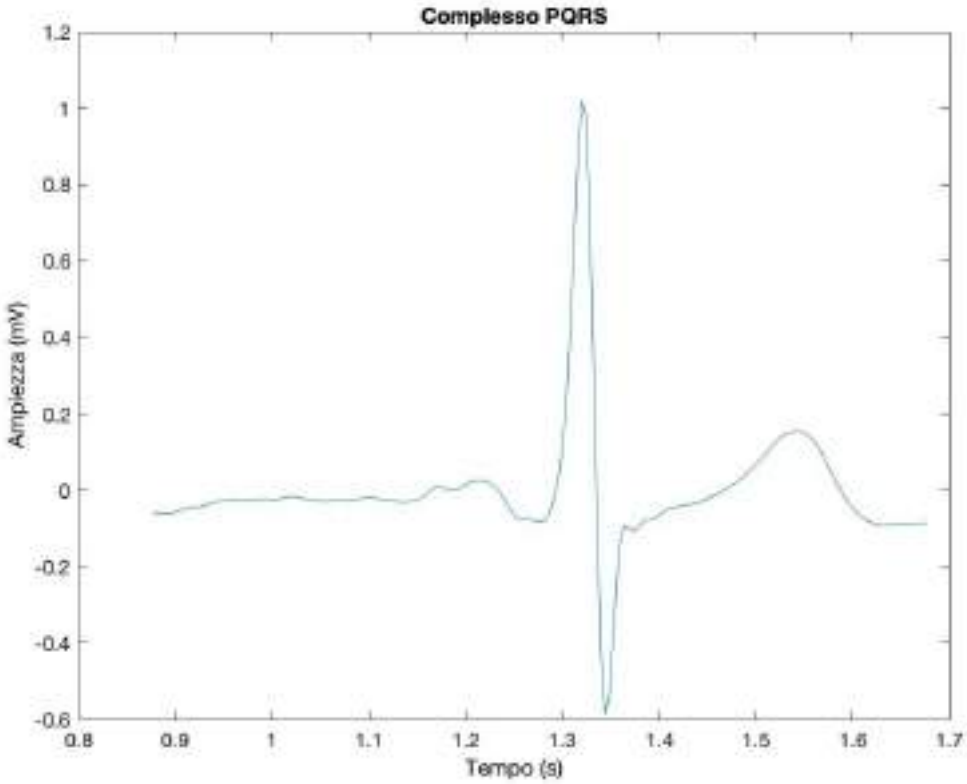
Il file DATA_Lab05_ECG.mat contiene il segnale ECG come vettore di 6500 campioni.

1. Caricare in Matlab il file DATA_Lab05_ECG.mat
2. Plottare il segnale ECG
3. Plottare in un'altra figura lo stesso segnale inserendo l'asse temporale. Considerare che la distanza temporale tra due campioni consecutivi (ovvero il periodo di campionamento del segnale reale) è di 0.004 s.
4. Calcolare il valore massimo del segnale ECG e l'indice dell'array a cui corrisponde (suggerimento: usare comando **max**)
5. Estrapolare e plottare un solo complesso PQRS (campione 220-420) dall'intero segnale ECG.
6. Salvare la figura come *complesso_PQRS.fig*

```m
%% Elaborazione Segnali Biomedici - Soluzione Laboratorio 05
% Prof. Veronese Mattia - UNIPD

%% OPENING
clear all
close all
clc

%% ESERCIZIO 1
% Caricare il file ECG.mat
load DATA_Lab05_ECG.mat

% Plot del segnale ECG
figure(1)
plot(ecg, 'LineWidth', 1.5)
xlabel('Campioni Misurati (#)');
ylabel('Ampiezza (mV)')
title('ECG')

% Definisco il vettore tempo associato al segnale
dt = 0.004;          % Periodo di campionamento
N = length(ecg);     % Totale campioni
fc = 1/dt;           % Frequenza di campionamento [Hz]
duration = N*dt;     % [Seconds] durata temporale del segnale

t = linspace(0,duration,N)';

% Plot del segnale ECG con asse temporale
figure(2)
plot(t,ecg, 'LineWidth', 1.5)
axis tight
xlabel('Tempo (s)')
ylabel('Ampiezza (mV)')
title('ECG')

% Valore massimo dell'ECG
[max_ecg, idx_max_ecg] = max(ecg);


% Estrapolare e rappresentare graficamente un complesso PQRS dal segnale ECG
pqrs = ecg(220:420,:);
t_pqrs = t(220:420,:);

% Plot del scomplesso PQRS
figure(3)
plot(t_pqrs, pqrs)
xlabel('Tempo (s)')
ylabel('Ampiezza (mV)')
title('Complesso PQRS')
savefig('complesso_PQRS.fig')
```

Soluzioni – Esercizio 1



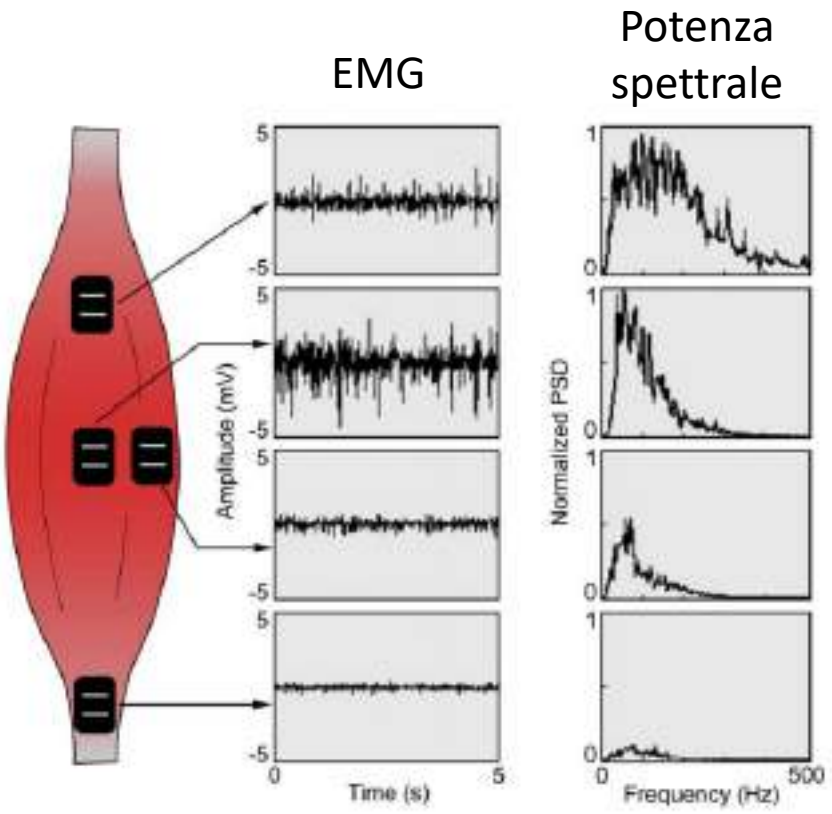


Il segnale EMG

È la registrazione dei potenziali elettrici che si formano in un muscolo durante la sua contrazione volontaria. Questi potenziali sono causati dalla depolarizzazione elettrica delle fibre muscolari in risposta all'arrivo di un impulso elettrico alla sinapsi neuromuscolare *(punto di contatto fra la terminazione di un nervo periferico e la membrana di una fibra musco)*

- Ampiezza: 1-10 mV
- Frequenza: 0-500 Hz

La contrazione ed espansione del muscolo scheletrico genera potenziali d'azione, che sono di importanza medica diagnostica.



Esercizio 2

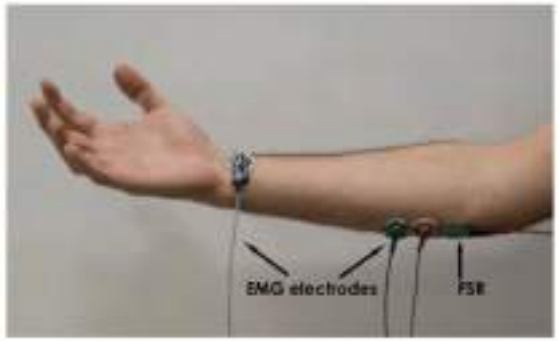
Il file DATA_Lab05_EMGforce.xlsx contiene il segnale elettrico EMG misurato da un trasduttore posizionato sull'avanbraccio destro di un soggetto sano. È stato chiesto al volontario di contrarre il trasduttore e rilassare il muscolo per cinque volte con forza crescente (dal 20% al 100% della contrazione volontaria massima).
Contenuto del file :

- Colonna 1: tempo in secondi, con frequenza di campionamento di 2000 Hz
- Colonna 2: forza in unità arbitraria
- Colonna 3: segnale EMG in mV

1) Caricare il file DATA_Lab05_EMGforce.xlsx, normalizza il segnale di forza in modo che il minimo sia zero e il massimo (corrispondente alla contrazione volontaria massima) sia 100. Fare il plot della forza e amplitudine del segnale EMG, e salvare la figura.

2) Estrapolare una porzione del segnale EMG che corrisponde ad una contrazione muscolare volontaria (ad esempio nell'intervallo temporale 17-21.48 secondi), e la corrispettiva forza. Plottare i due segnali in due riquadri separati (suggerimento: usare subplot)

3) Data una soglia di forza di 60 [%MVC], trova dove il segnale è maggiore della soglia (suggerimento: usare il comando find). Plottare il segnale originale di forza indicando la soglia con una linea orizzontale, e il segnale di forza sogliato, in riquadri diversi.

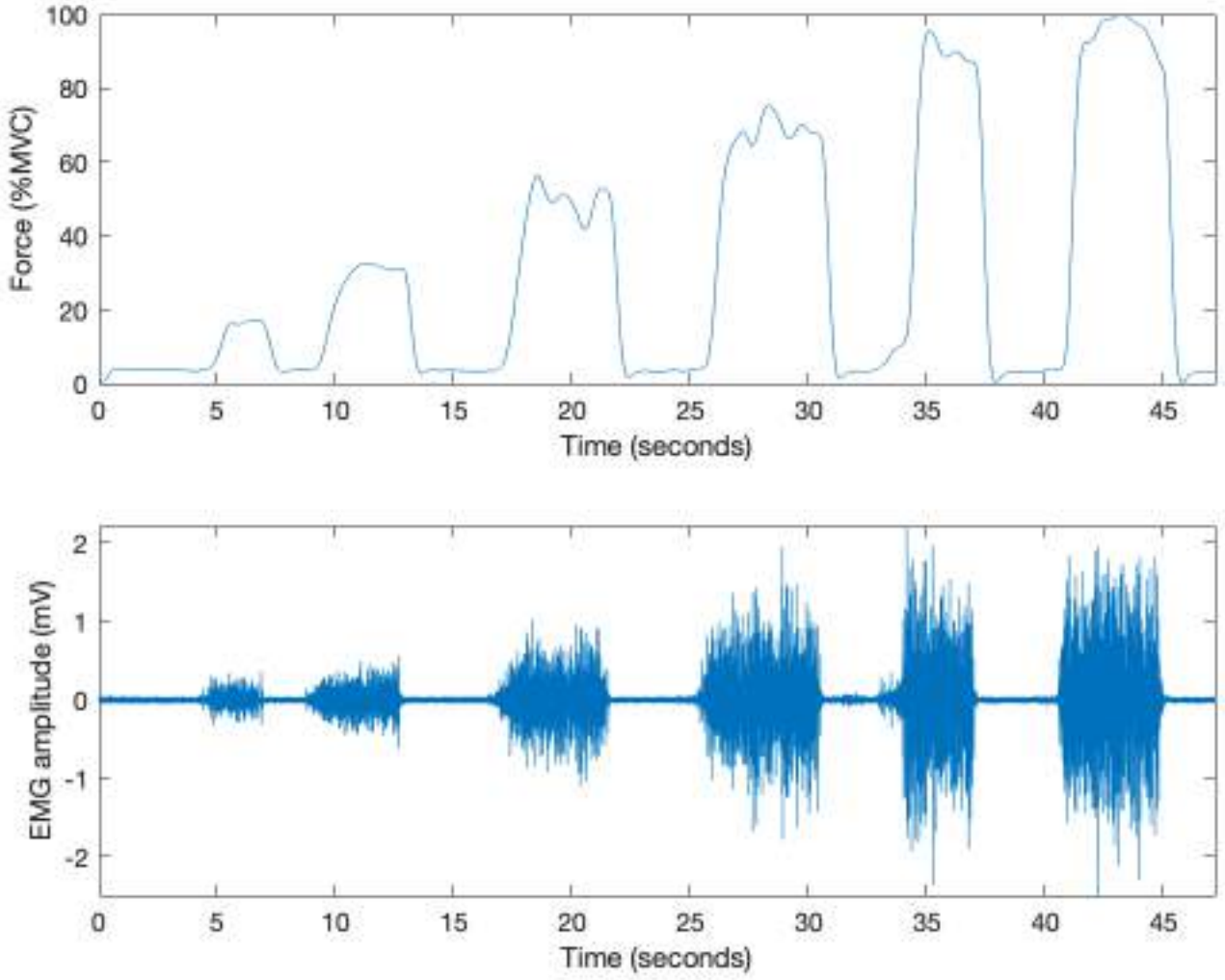


Elettrodi EMG

```m
%5 Elaborazione Segnali Biomedici - Soluzione Laboratorio 05
% Prof. Veronese Mattia - UNIPD

%% OPENING
clear all
close all
clc


%% Esercizio 2
% Caricare il file contenente il segnale EMG
data = xlsread('DATA_Lab05_EMGforce.xlsx');

time = data(:,1); % Time in seconds with sampling rate = 2000 Hz
force = data(:,2); % Force in arbitrary units
emg = data(:,3); % EMG signal in mV

% Normalizzare il segnale EMG in modo che il minimo sia 0 e il massimo sia
% 100
force_norm= 100*(force-min(force))/(max(force)-min(force));

%% 1) Rappresentare la forza e amplitudine del segnale EMG
figure(1);
subplot(2,1,1)
plot(time, force_norm)
axis tight
xlabel('Time (seconds)')
ylabel('Force (%MVC)')
subplot(2,1,2)
plot(time, emg)
axis tight
xlabel('Time (seconds)')
ylabel('EMG amplitude (mV)')
axis tight;

%% 2) Estrapolare una porzione del segnale EMG che corrisponde ad una
% contrazione muscolare volontaria (ad esempio nell'intervallo temporale
% 17-21.48 s) e la corrispettiva forza. Plottare i due segnali in due
% riquadri separati.
time_start_contraction = 17;    % Seconds
time_end_contraction = 21.48;   % Seconds
contraction_force = find((time>time_start_contraction)&(time<time_end_contraction));

figure(2)
subplot(2,1,1)
plot(time(contraction_force),force_norm(contraction_force))
axis tight
xlabel('Time (seconds)')
ylabel('Force (%MVC)')
subplot(2,1,2)
plot(time(contraction_force),emg(contraction_force))
axis tight
xlabel('Time (seconds)')
ylabel('EMG amplitude (mV)')
axis tight
savefig('contrazione_muscolare.fig')

%% 3) Data una soglia di forza di 60 [%MVC], trova dove il segnale e'
% maggiore della soglia. Plottare il segnale originale di forza indicando
% la soglia con una linea orizzontale, e il segnale di forza "sogliato", in
% riquadri diversi
threshold_force = 60;
idx = find(force_norm>threshold_force);
force_thresholded = force_norm;

force_thresholded(idx) = threshold_force;

figure(3);
subplot(2,1,1)
plot(time, force_norm)
hold on
yline(threshold_force, 'r')
axis tight
xlabel('Time (seconds)')
ylabel('Force (%MVC)')
title('Original force signal')
subplot(2,1,2)
plot(time, force_thresholded)
axis tight
xlabel('Time (seconds)')
ylabel('Force (%MVC)')
title('Thresholded force signal')
```

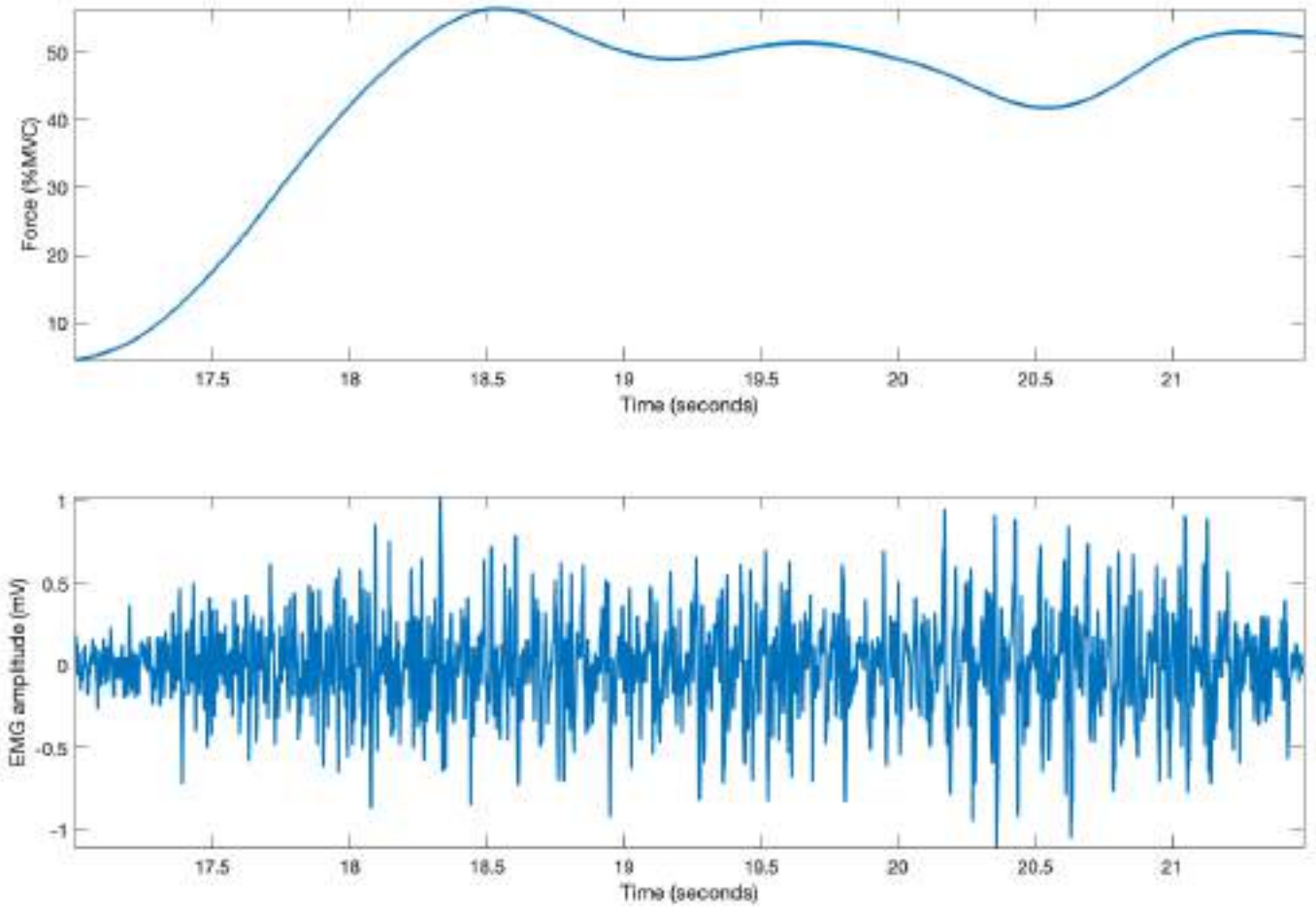
Soluzioni – Esercizio 2

1)



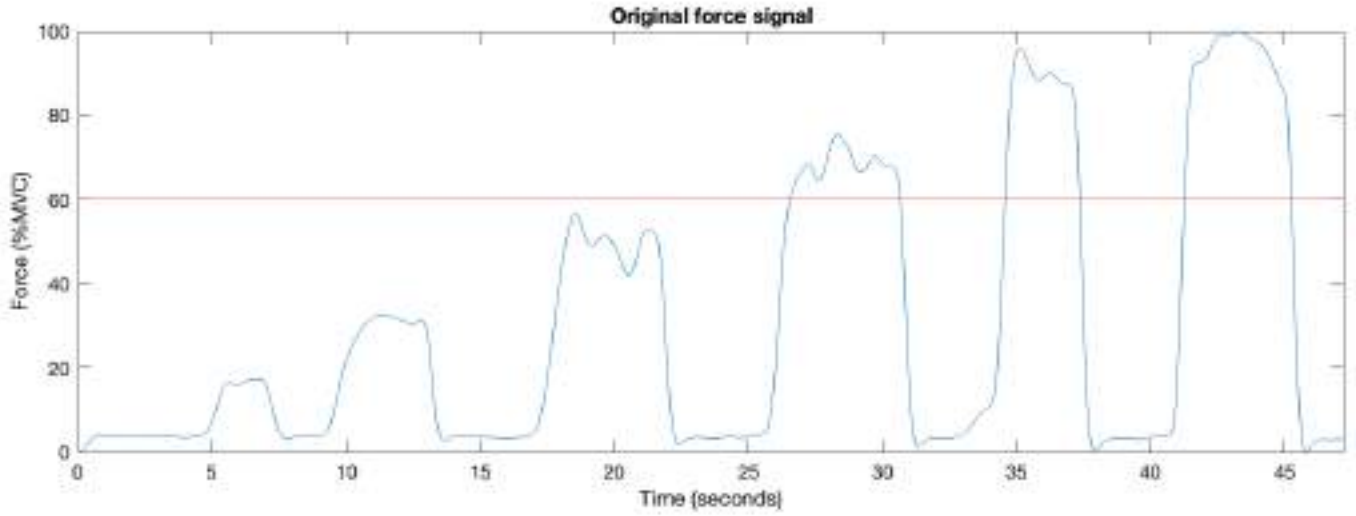
Soluzioni – Esercizio 2

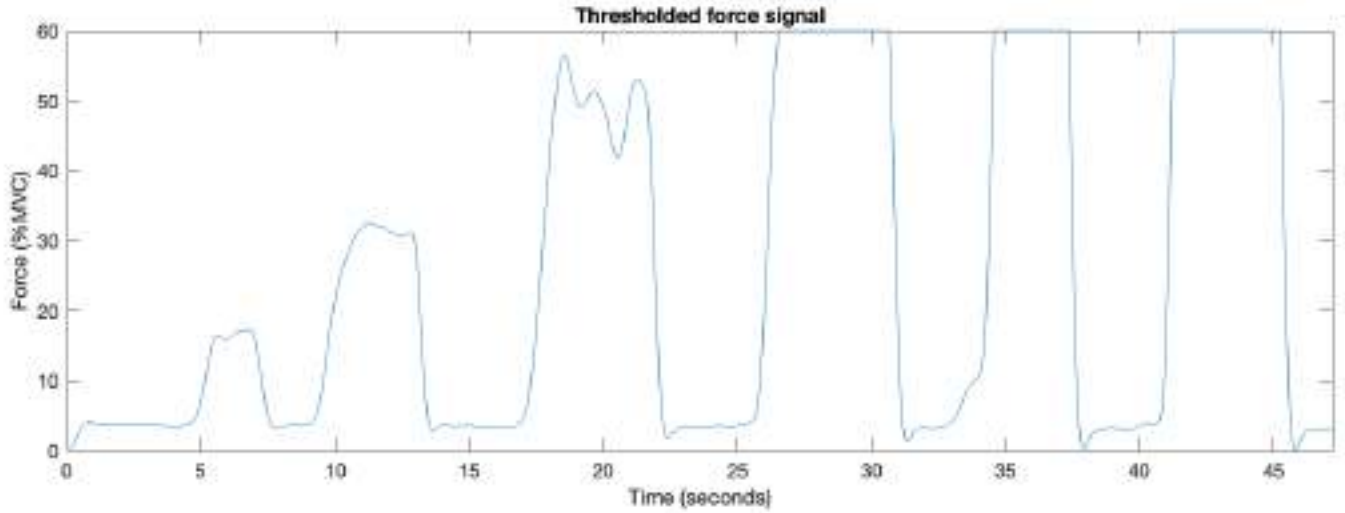
2)



Soluzioni – Esercizio 2

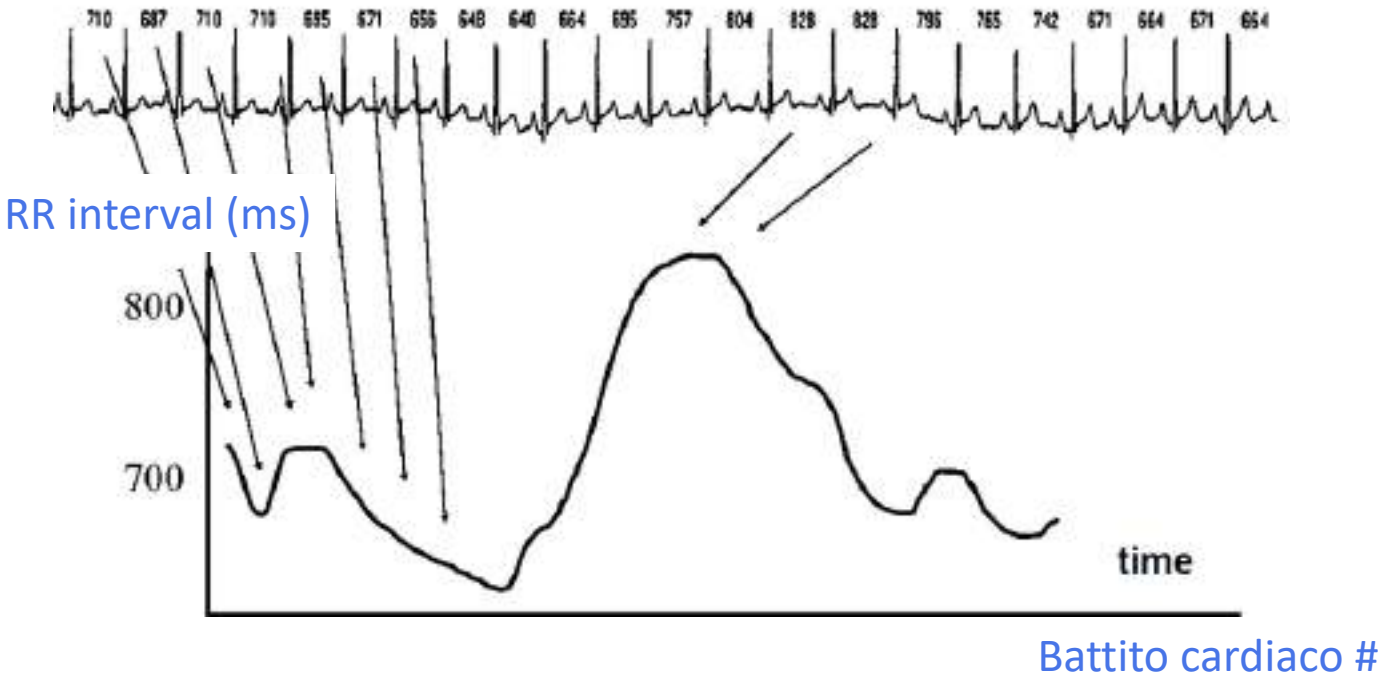
3)





Periodo e frequenza di un segnale ECG

Il tacogramma fornisce una misura della variazione nel tempo del ciclo cardiaco. Il diagramma del tacogramma ha sull'asse delle x il numero dei battiti cardiaci, e sull'asse delle y il valore dell'intervallo R-R



RIPASSO - La principale unità di misura per la frequenza di ECG è *battiti per minuto* (bpm). bpm è definita come frequenza media relativa ($n_{battiti}$) ad un intervallo di tempo (t, min)

$$Bpm = 1/T*60$$

Esercizio 3

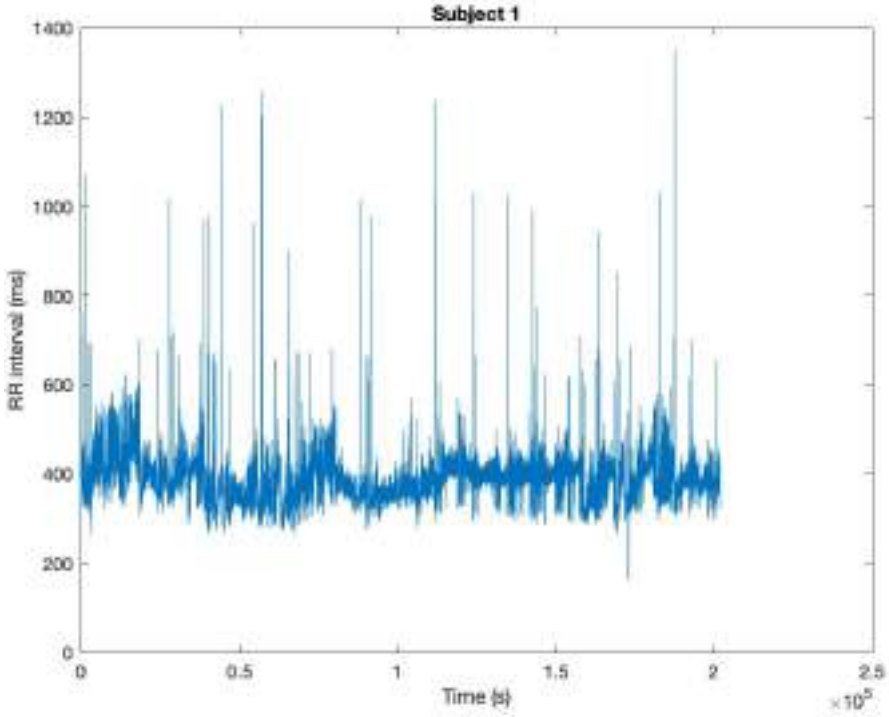
I file DATA_Lab05_Subject1.txt e DATA_Lab05_Subject2.txt contengono gli intervalli RR di due soggetti sani misurati con il sistema di monitoraggio Holter per 24h.

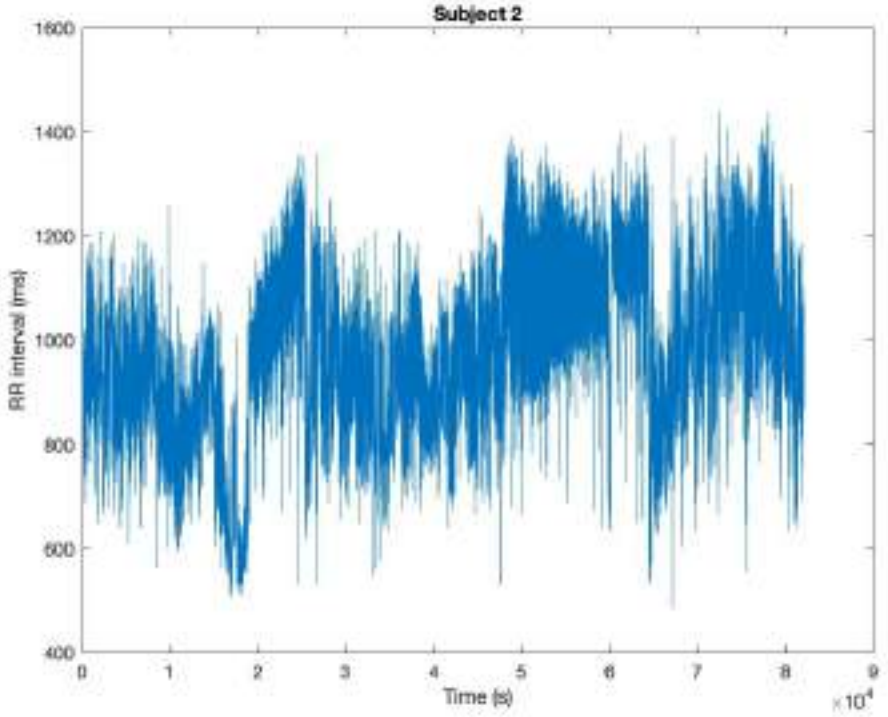
	Subject 1	Subject 2
Age (years)	0.17	55
Gender	M	M

1) Caricare i file in Matlab
2) Plottare il segnale RR in funzione del tempo
3) Calcolare la frequenza cardiaca e i battiti per minuto (bpm)
4) Calcolare il battito cardiaco medio (verificare se ci sono outliers da escludere)

```m
%% Elaborazione Segnali Biomedici - Soluzione Laboratorio 05
% Prof. Veronese Mattia - UNIPD

%% OPENING
clear all
close all
clc


%% Esercizio 1: RR interval time series from two healthy subjects - Holter
% monitoring for 24h
% Dataset available at https://physionet.org/content/rr-interval-healthy-subjects/1.0.0/

%% Primo soggetto (ID: 4040)
data_Sbj1 = importdata('DATA_Lab05_Subject1.txt');

figure(1)
plot(data_Sbj1)
axis tight
xlabel('Numero di battiti cardiaci')
ylabel('Intervallo RR (ms)')
title('Soggetto ID 4040')

%% Calcolare i battiti per minuto e la frequenza cardiaca per ogni intervall RR

bpm_Sbj1 = zeros(size(data_Sbj1));   % Battiti per minuto

F_sbj1 = zeros(size(data_Sbj1));     % Frequenza cardiaca

for i=1:size(data_Sbj1,1)
   F_sbj1(i) = 1/(data_Sbj1(i)/1000);
   bpm_Sbj1(i) = F_sbj1(i)*60;
end

% Battito cardiaco medio
mean_HR_Sbj1 = mean(bpm_Sbj1);


%% Secondo soggetto (ID: 10)
data_Sbj2 = importdata('DATA_Lab05_Subject2.txt');

figure(2)
plot(data_Sbj2)
axis tight
xlabel('Numero di battiti cardiaci')
ylabel('Intervallo RR (ms)')
title('Soggetto ID 10')

% Calcolare i battiti per minuto e la frequenza cardiaca per ogni intervall RR
bpm_Sbj2 = zeros(size(data_Sbj2));   % Battiti per minuto

F_sbj2 = zeros(size(data_Sbj2));     % Frequenza cardiaca

for i=1:size(data_Sbj2,1)
   F_sbj2(i) = 1/(data_Sbj2(i)/1000);
   bpm_Sbj2(i) = F_sbj2(i)*60;
end

% Battito cardiaco medio
mean_HR_Sbj2 = mean(bpm_Sbj2);
```

Soluzioni – Esercizio 3



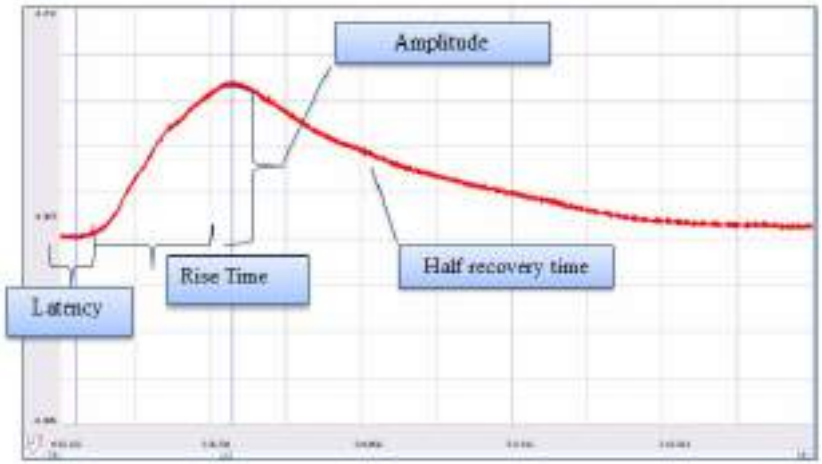


Conduttanza cutanea / Risposta galvanica della pelle (GSR)

Il **GSR** è la misura delle variazioni continue nelle caratteristiche elettriche della pelle, come ad esempio la conduttanza, a seguito della variazione della sudorazione del corpo umano.

Per misurare il segnale GSR si usano due elettrodi applicati al dito indice e medio di una mano. La variazione di una corrente a basso voltaggio applicate tra i due elettrodi è utilizzata come misura dell'attività elettrodermica (EDA).

Evidenziate relazioni tra segnale GSR e alcuni stati mentali (stress, stanchezza, coinvolgimento).



Esercizio 4

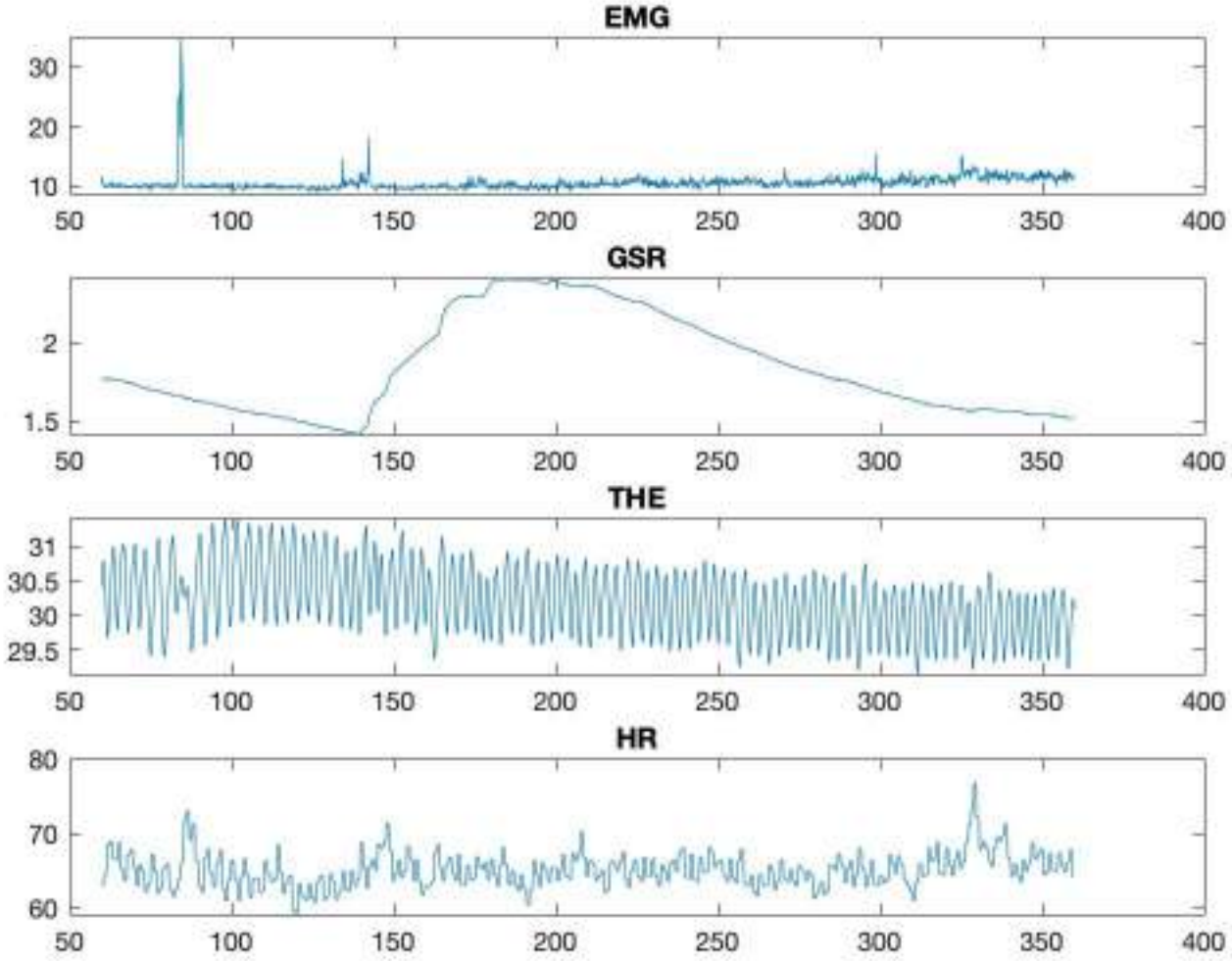
Il file DATA_Lab05_TestFisiologico.xlsx contiene i seguenti segnali registrati in un soggetto:

- Colonna 1: Tempo di campionamento (Time)
- Colonna 2: Attività mioelettrica (EMG)
- Colonna 3: Conduttanza cutanea (GSR)
- Colonna 4: Temperatura periferica (THE)
- Colonna 5: Frequenza cardiaca (HR)

1) Plottare in un'unica figura, in finestre diverse, i segnali EMG, GSR, THE, HR in funzione del tempo
2) Creare una struttura (usare il comando *struct*) di dimensioni 4x1 denominata **segnali**, dove ogni elemento contiene informazioni di uno solo dei segnali visti. In particolare ogni elemento della struttura deve avere i seguenti campi:
   - nome_segnale che deve valere EMG, GSR, HR o THE
   - dati che deve contenere il vettore dei dati
   - tempo che deve contenere il vettore tempo
   - stat che è un cell array di dimensioni 2x1 che deve contenere un vettore con i valori minimo e Massimo del segnale uno scalare che rappresenta la deviazione standard.

```
% Elaborazione Segnali Biomedici - Soluzione Laboratorio 05
% Prof. Veronese Mattia - UNIPD

%% OPENING
clear all
close all
clc

%% Esercizio 3

% Caricare il file DATA_Lab06_TestFisiologico.xlsx
data = xlsread('DATA_Lab05_TestFisiologico.xlsx');

label_signali = ["EMG", "GSR", "THE", "HR"];
tot_signali = size(data,2)-1;

figure
% Plot dei segnali
for i=1:tot_signali

    subplot(tot_signali,1,i)
    plot(data(:,1),data(:,i+1))
    title(label_signali(i))

end % for

% Creare una struttura array per i signali, che contenga i nomi dei
% segnali, i dati dei segnali, il vettore tempo e un cell array 'stat' 1x2
% contenente una matrice con il valore minimo e massimo di ciascun segnale,
% e un vettore con la standard deviation di ciascun segnale
segnali = struct('nome_segnale', label_signali);

segnali.dati = data(:,2:tot_signali+1);
segnali.tempo = data(:,1);

max_min_signali = zeros(tot_signali,2);
std_signali = zeros(tot_signali,1);

for i=1:tot_signali

    max_min_signali(i,1) = max(data(:,i+1));
    max_min_signali(i,2) = min(data(:,i+1));
    std_signali(i,1) = std(data(:,i+1));

end % for

segnali.stat = {max_min_signali, std_signali};
```

Soluzioni – Esercizio 4

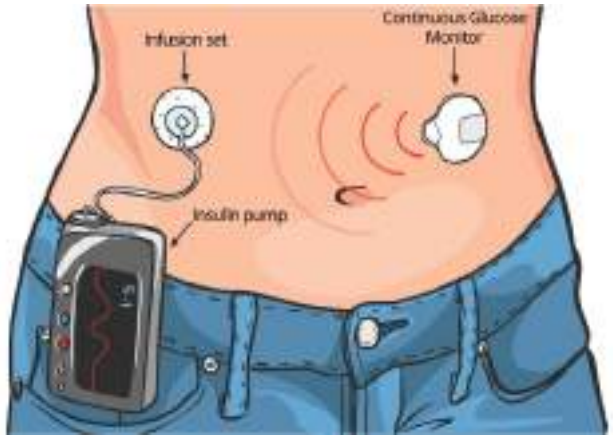


Esercizio per casa

Il file DATA_Lab05_Es5.mat contiene il segnale glicemico (CGM: continuous glucose monitoring) misurato su un paziente con diabete di tipo I, insieme ai dati demografici e le informazioni delle iniezioni di insulina dalla pompa insulinica.
Le informazioni contenute nel file .mat sono descritte in dettaglio nel file Data_Lab05_Es5.xlsx. Il tempo di campionamento del segnale glicemico è di 5 minuti.

CONSEGNA ESERCIZIO:

1) Caricare il file DATA_Lab05_Es5.mat
2) Considerando come valori normali di glucosio quelli compresi tra 70-250 mg/dl, plottare il segnale CGM in funzione del tempo e indicare la soglia di valori normali con due line rosse. (Suggerimento: usare il comando **linspace** per creare il vettore tempo)
3) Identificare gli eventi di ipoglicemia ed iperglicemia e calcolare la loro durata nel corso di tutto il tracciato (suggerimento: usare il comando **find**) .



```m
%% Elaborazione Segnali Biomedici - Soluzione Laboratorio 04
% Prof. Veronese Mattia - UNIPD

%% OPENING
clear all
close all
clc

%% Esercizio 3
% 1) Caricare il file contenente i dati
load DATA_Lab05_es5.mat

%% 2) Rappresentare il segnale glicemico in funzione del tempo,
% indicando la soglia di valori normali (70-250 mg/dl) con due linee rosse

low_level_glucose = 70;       % mg/dl
upper_level_glucose = 250;    % mg/dl

% Durata di acquisizione del segnale glicemico
t_c = 5;    % Tempo di campionamento [minuti]
CGM_values = CGM.value;
N=length(CGM_values); % Numero di campioni
duration = t_c*N/60;    % Tempo totale di acquisizione [ore]
time = linspace(0, duration, N);

figure(1)
plot(time, CGM_values )
yline(upper_level_glucose,'-r')
yline(low_level_glucose,'-r')
xlabel('Time (hours)')
ylabel('Glucose (mg/dl)')
title('CGM signal')


%% 3) Identificare gli eventi di ipoglicemia ed iperglicemia e
% calcolare la loro durata nel corso di tutto il tracciato

% Indici eventi ipoglicemici e iperglicemici
idx_eventi_ipoglicemici = find(CGM_values <low_level_glucose);
idx_eventi_iperglicemici = find(CGM_values >upper_level_glucose);

% Corrispettivi tempi degli eventi ipoglicemici e iperglicemici
t_eventi_ipoglicemici = time(idx_eventi_ipoglicemici);
t_eventi_iperglicemici = time (idx_eventi_iperglicemici);

% Calcola la differenza tra gli indici successivi
diff_ipoglicemia = diff(idx_eventi_ipoglicemici);
diff_iperglicemia = diff(idx_eventi_iperglicemici);

% Trova gli indici degli eventi che terminano
idx_end_eventi_ipoglicemia = find(diff_ipoglicemia > 1);
idx_end_eventi_iperglicemia = find(diff_iperglicemia > 1);

% Numero totale di eventi ipoglicemici e iperglicemici
N_eventi_ipoglicemia = length(idx_end_eventi_ipoglicemia)+1; % +1 per quello finale
N_eventi_iperglicemia = length(idx_end_eventi_iperglicemia)+1;

% Vettori dove inserire la durata degli eventi ipoglicemici e iperglicemici
durata_eventi_ipoglicemia = zeros(N_eventi_ipoglicemia,1);
durata_eventi_iperglicemia = zeros(N_eventi_iperglicemia,1);

% Calcolo della durata degli eventi ipoglicemici
start = 1;
for i=1:N_eventi_ipoglicemia

    if i == N_eventi_ipoglicemia
            durata_eventi_ipoglicemia(i,:) = t_eventi_ipoglicemici(end)- t_eventi_ipoglicemici(start);
    else
        durata_eventi_ipoglicemia(i,:) = t_eventi_ipoglicemici(idx_end_eventi_ipoglicemia(i))-t_eventi_ipoglicemici(start);
        start = idx_end_eventi_ipoglicemia(i) + 1;
    end % end if

end % for

% Calcolo della durata degli eventi iperglicemici
start = 1;
for i=1:N_eventi_iperglicemia

    if i == N_eventi_iperglicemia
            durata_eventi_iperglicemia(i,:) = t_eventi_iperglicemici(end)- t_eventi_iperglicemici(start);
    else
        durata_eventi_iperglicemia(i,:) = t_eventi_iperglicemici(idx_end_eventi_iperglicemia(i))-t_eventi_iperglicemici(start);
        start = idx_end_eventi_iperglicemia(i) + 1;
    end % end if

end % for
```

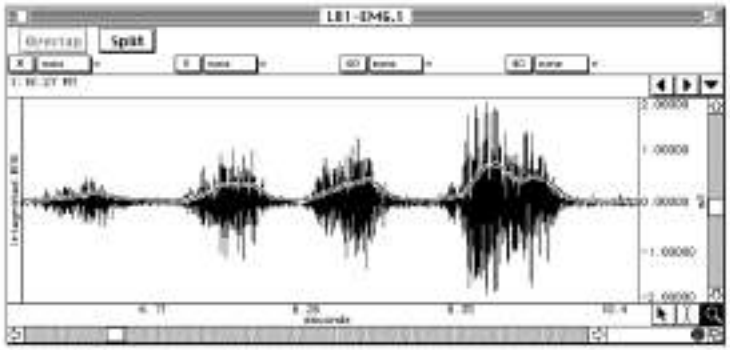
Analisi EMG nella contrazione muscolare delle braccia

Tutor: Dr. Giovanna Nordio e Giulia Vallini

Prof. Mattia Veronese
Email: mattia.veronese@unipd.it
Dipartimento di Ingegneria dell'Informazione

Ricevimento: su appuntamento (e-mail)
Edificio DEI/A, piano 2o, stanza 214

Segnale EMG misurato con un sistema di acquisizione BIOPAC



BIOPAC Systems, Inc.
42 Aero Camino, Goleta, CA 93117 (805) 685-0066, Fax (805) 685-0067 Email: info@biopac.com
Web Site: http://www.biopac.com

Esercizio

Il sistema BIPOPAC è stato utilizzato per misurare il segnale EMG del braccio dominante e non dominante di un volontario. E' stato chiesto al volontario di contrarre, ogni 2 secondi per 2 secondi, volontariamente il braccio con forza crescente per 4 volte, ripentendo poi la contrazione con forza massima per ulteriori 4 volte, per un totale di 8 contrazioni per braccio.
La figura 1 rappresenta un esempio di segnale ottenuto con il sistema BIOPAC.

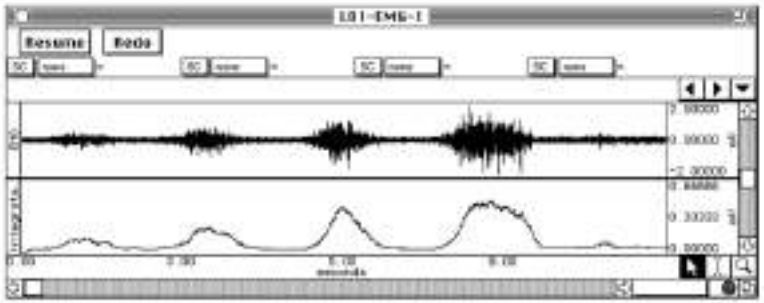


Figura 1

1) Caricare il file **EMG_bothArm_crescente.mat** in Matlab e plottare i segnali come in Figura 1 nella stessa figura (suggerimento: usare comando **subplot**). Il file contiene le seguenti variabili:

   data → prima colonna *EMG*, seconda colonna *integrated EMG*
   isi → inter stimulus interval (periodo di campionamento)
   isi_units → unità di misura del periodo di campionamento
   units → unità di misura del segnale EMG e integrated EMG

Esercizio

2) Calcolare il valore medio, massimo, minimo e peak-to-peak (P-P) per ogni singola contrazione del braccio dominante (*Suggerimento: ricavate i tempi di ogni contrazione usando il cursore sulla Figura 1*). Ripetere il calcolo poi anche per il braccio non dominante.

P-P in una contrazione: v_max– v_min

3) Per ogni parametro rilevato (mean, min, max, P-P) per le 5 misurazioni a forza massima calcolare media e SD.

4) Per ogni parametro rilevato (mean, min, max, P-P) confrontare la media ottenuta nelle 5 misurazioni a forza massima (punto 3) con i corrispettivi valori di dati ottenuti a forza minima e determinare l'aumento percentuale per il braccio dominante e non dominante

Aumento_% = [(v_max-v_min)/v_min*100]

Esercizio [punto 2]

Braccio dominante

	Media	Max	Min	P-P
Contrazione_1				
…				
Contrazione_8				

Braccio non dominante

	Media	Max	Min	P-P
Contrazione_1				
…				
Contrazione_8				

Esercizio [punto 3-4]

FORZA MASSIMA

Braccio dominante	**Media**	**Max**	**Min**	**P-P**
Mean				
SD				

Braccio non dominante	**Media**	**Max**	**Min**	**P-P**
Mean				
SD				

FORZA MASSIMA vs FORZA MINIMA

Braccio dominante	**Media**	**Max**	**Min**	**P-P**
Aumento_%				

Braccio non dominante	**Media**	**Max**	**Min**	**P-P**
Aumento_%				

```
clc% Elaborazione Segnali Biomedici - Soluzione Laboratorio 07
% Prof. Veronese Mattia - UNIPD

%% OPENING
clear all
close all
clc

%% Esercizio BIOPAC

% 1) Caricare e plottare il segnale EMG
load EMG_bothArm_crescente.mat

dt = 0.002;                                 % Tempo di campionamento [s]
N_campioni = size(data,1);                  % Numero totale di campioni
duration = (N_campioni-1)*dt;                 % Durata di acquisizione del segnale
time = linspace(0,duration,N_campioni);   % Vettore tempo

figure(1)
subplot(2,1,1)
% plot(time, data(:,1))
plot(data(:,1))
xlabel('Time [s]')
ylabel('mV')
title('EMG')
subplot(2,1,2)
plot(time, data(:,2))
xlabel('Time s[]')
ylabel('mV')
title('Integrated EMG')

%% 2) Creare matrice contenente, valore medio, valore massimo, valore minimio e picco-to-picco per ogni singola contrazione del braccio domte
ind_contractions_arm1 = [979,2001;2951,4051;4993,5993;6922,8022;8887,10087;11090,11990;12890,13990;15010,16210];

tot_contractions_arm1 = size(ind_contractions_arm1, 1);
stats_arm1 = zeros(tot_contractions_arm1, 4);

for i=1:tot_contractions_arm1
    contractions_arm1 = data(ind_contractions_arm1(i,1):ind_contractions_arm1(i,2),1);
    stats_arm1(i, 1) = mean(contractions_arm1);
    stats_arm1(i, 2) = max(contractions_arm1);
    stats_arm1(i, 3) = min(contractions_arm1);
    stats_arm1(i, 4) = stats_arm1(i, 2) - stats_arm1(i, 3);

end     % for

% Creare matrice contenente, valore medio, valore massimo, valore minimio e picco-to-picco per ogni singola contrazione del braccio NON dominante
times_contractions_arm2 = [17980,18930;19960,21000;21980,22980;23970,24980;25940,27030;27940,29080;30010,30910;31940,32900];

tot_contractions_2 = size(times_contractions_arm2, 1);
stats_arm2 = zeros(tot_contractions_2, 4);

for i=1:tot_contractions_2
    contractions_arm2 = data(times_contractions_arm2(i,1):times_contractions_arm2(i,2),1);
    stats_arm2(i, 1) = mean(contractions_arm2);
    stats_arm2(i, 2) = max(contractions_arm2);
    stats_arm2(i, 3) = min(contractions_arm2);
    stats_arm2(i, 4) = stats_arm2(i, 2) - stats_arm2(i, 3);
end     % for

disp('VERIFICA LE VARIABILI stats_arm* PER LE SOLUZIONI ESERCIZIO - PARTE 2')

%% 3) Per ogni parametro rilevato (mean, max, …) per le 5 misurazioni a forza massima calcolare media e SD
stats_arm1_max_contractions = zeros(2, 4);
stats_arm2_max_contractions = zeros(2, 4);

for i=1:4
    % Braccio dominante
    stats_arm1_max_contractions(1,i) = mean(stats_arm1(4:end,i));
    stats_arm1_max_contractions(2,i) = std(stats_arm1(4:end,i));
    % Braccio non dominante
    stats_arm2_max_contractions(1,i) = mean(stats_arm2(4:end,i));
    stats_arm2_max_contractions(2,i) = std(stats_arm2(4:end,i));

end


% 4) Per ogni parametro rilevato (mean, min, max, P-P) confrontare la media ottenuta
% al passo precedente con il valore relativo al gruppo di dati a forza minima
% e determinare l'aumento percentuale nelle due diverse situazioni
% Aumento percentuale del braccio dominante
aumento_perc_arm1 = zeros(1,4);
for i=1:size(aumento_perc_arm1, 2)
    aumento_perc_arm1(:,i) = (stats_arm1_max_contractions(1,i)-stats_arm1(1,i))/stats_arm1(1,i)*100;
end     % for

% Aumento percentuale del braccio NON dominante
aumento_perc_arm2 = zeros(1,4);
for i=1:size(aumento_perc_arm2, 2)
    aumento_perc_arm2(:,i) = (stats_arm2_max_contractions(1,i)-stats_arm2(1,i))/stats_arm2(1,i)*100;
end     % for

disp('VERIFICA LE VARIABILI aumento_perc_arm* PER LE SOLUZIONI ESERCIZIO - PARTE 3')

```

Conversione A/D

Tutor: Dr. Giovanna Nordio e Giulia Vallini

Prof. Mattia Veronese
Email: mattia.veronese@unipd.it
Dipartimento di Ingegneria dell'Informazione

Ricevimento: su appuntamento (e-mail)
Edificio DEI/A, piano 2o, stanza 214

Obiettivi del laboratorio

- **Quantizzare un segnale dato un certo numero possibile di livelli e una strategia di approssimazione (arrotondamento o troncamento)**

- **Studiare come diverse tipi di interpolazioni permettono di approssimare l'andamento dei segnali nel tempo**

Esercizio 1 – CHALLENGE

Si consideri un segnale sinusoidale con ampiezza di 1.5 V e un periodo di oscillazione di 100 secondi, campionato con una frequenza di 1Hz

1) Rappresentare graficamente il segnale sinusoidale nell' intervallo temporale da 0 a 150 secondi.

2) Definire due quantizzatori associati ad una codifica con segno a 4 bit, uno per arrotondamento (suggerimento: usare il comando **round**) e uno per troncamento (suggerimento: usare il comando **fix**), e calcolare il corrispondente errore di quantizzazione.

3) Visualizzare il segnale sinusoidale originale, e i due segnali quantizzati in un' unica figura.

Esercizio 1 – ISTRUZIONI DI CONSEGNA [Per prepararsi all'esame]

Consegnare un file .mat sul sito moodle del corso (sezione LABORATORIO) contenente:

- **Una struttura *'quant'* che contenga:**
  - **quantizzatore ad arrotondamento (quant(1).data)**
  - **errore del quantizzatore ad arrotondamento (quant(1).err)**
  - **quantizzatore a troncamento (quant(2).data)**
  - **errore del quantizzatore a troncamento (quant(2).err)**

Attenzione:

- Matlab è case sensitive (sbagliare il nome delle variabili equivale a sbagliare l'esercizio)

```m
% Elaborazione Segnali Biomedici - Soluzione Laboratorio 11
% Prof. Veronese Mattia - UNIPD


%% OPENING
clear all
close all
clc

%% Definizione del segnale sinusoidale

A = 1.5;                % Ampiezza [V]
T = 100;                % Periodo del segnale [s]
Fc = 1;                 % Frequenza di campionamento [Hz]
time = [0:1/Fc:150];      % Time [s]

% Segnale sinusoidale
y = A*sin(2*pi*1/T*time);

%% Definizione del quantizzatore ad arrotondamento e troncamento

n_bit = 4;                          % Numero di bit
range_y = [min(y), max(y)];         % Range del segnale sinusoidale
N_livelli = 2^n_bit-1;              % Numero di livelli per la codifica
q = max(abs(range_y))*2/N_livelli;          % Passo di quantizzazione - ATTENZIONE PERCH◆ CODIFICA CON SEGNO ◆ SIMMETRICA

% PROBLEMA: IL NUMERO DI LIVELLI DI yq_arr ◆ effettivamente superiore a N_livelli

% Quantizzatore ad arrotondamento
% yq_arr = q * round(y/q);
yq_arr = q * sign(y).*min(abs(round(y/q)),(N_livelli-1)/2);
% Errore di quantizzazione
e_arr = yq_arr - y;

% Quantizzatore a troncamento
% yq_trunc = q * fix(y/q);
yq_trunc = q * sign(y).*min(abs(fix(y/q)),(N_livelli-1)/2);

% Errore di quantizzazione
e_trunc = yq_trunc - y;

% Rappresentazione del segnali sinusoidale originale e dei segnali
% quantizzati sovrapposti
figure
plot(time,y)
xlabel('Time [sec]')
ylabel('Segnale')
hold on
stem(time,yq_arr, 'rx')
stem(time,yq_trunc, 'g.')

% Salvataggio dei risultati in una struttura
quant(1).data = yq_arr;
quant(1).err = e_arr;
quant(2).data = yq_trunc;
quant(2).err = e_trunc;

save Resultati_Lab11_esercizio_1 quant


% Rappresentazione degli errori di arrotondamento
figure
plot(time,e_arr, 'r')
hold on
plot(time,e_trunc, 'g')
xlabel('Time [sec]')
ylabel('QUANTIZ. ERROR')
```

Interpolazione in Matlab

Interpolazione in Matlab è spesso implementata tramite il comando **interp1**

yi = interp1(x, y, xi, metodo)

che restituisce un vettore **yi** i cui elementi sono i valori delle ordinate interpolate ottenute in corrispondenza dell'ascissa **xi**, note le ascisse **x** e le ordinate **y** che concorrono a definire la curva interpolante.

Metodo di interpolazione:
- *linear*: interpolazione tramite spline cubiche
- *spline*: interpolazione tramite spline cubiche
- *cubic*: interpolazione con cubiche di Hermite
- *pchip*: interpolazione con cubiche di Hermite
- *nearest*: interpolazione al punto più vicino

Attenzione. Il vettore **xi** deve essere contenuto nel vettore **x**, altrimenti non si parla più di interpolazione ma di estrapolazione del segnale.

Esercizio 2

Data la funzione

$$y = 1/(1+x^2)$$

nota nei punti x = 5 cos(z), con z linearmente distribuito tra 0 e π in 11 campioni

1) Disegnare il valore della funzione in questi punti
2) Trovare una approssimazione della funzione in 100 punti equispaziati (suggerimento: usare il comando ***linspace***) dell'intervallo [−5, 5], usando l'interpolazione lineare e spline (usare il comando ***interp1***).
   Plottare sulla stessa figura (usare il comando ***hold on***) del punto 1) le due interpolazioni.

MATLAB Drive > Lab10_Es2_9ddd399c73a8444372b2919b155e7984.m

```
% Elaborazione Segnali Biomedici - Soluzione Laboratorio 10
% Prof. Veronese Mattia - UNIPD

%% OPENING
clear all
close all
clc


%% Esercizio 2

%Creazione del segnale
n=11;
z=linspace(0, pi, n);
x = 5*cos(z) ;
y = 1./(1+ x.^2);

%Plot data
figure
plot(x ,y ,'og')

%Calcolo interpolazione
m = 100;
interpx = linspace(-5, 5 , m);
interpy_linear= interp1 (x, y, interpx, 'linear');
interpy_spline= interp1 (x, y, interpx, 'spline');

hold on
plot (interpx, interpy_linear, 'r')
plot (interpx, interpy_spline, 'b')
legend ('dati', 'Interpolazione Lineare','Interpolazione Spline')
xlabel('x (u.a.)')
ylabel('y (u.a.)')
```

Potenziali Evocati e rumore

Tutor: Dr. Giovanna Nordio e Giulia Vallini

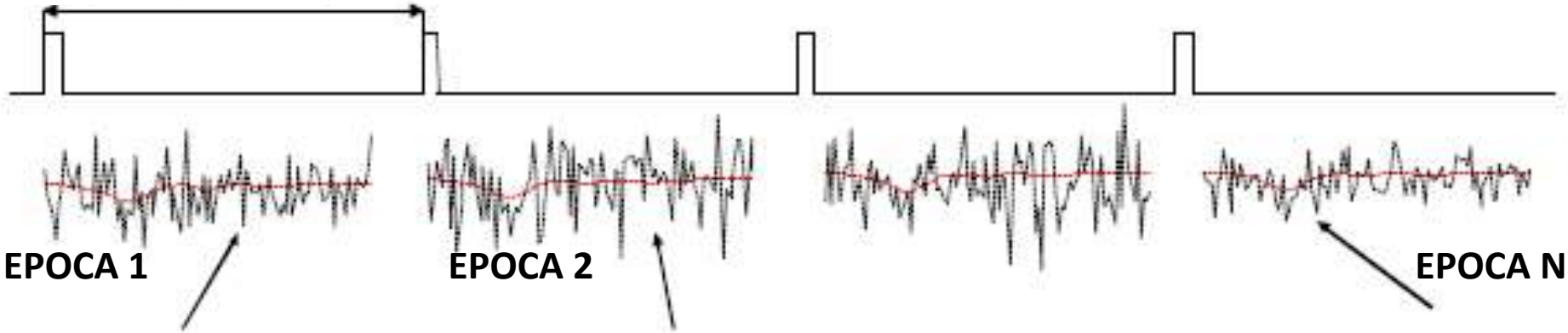
Prof. Mattia Veronese
Email: mattia.veronese@unipd.it
Dipartimento di Ingegneria dell'Informazione

Ricevimento: su appuntamento (e-mail)
Edificio DEI/A, piano 1o, stanza 106

Potenziali evocati e rumore

Al soggetto vengono proposti N stimoli identici ed equispaziati nel tempo. Dopo ogni stimolo viene misurata una *epoca* del segnale, che contiene sia potenziale evocato che rumore elettroencefalografico (rumore additivo).

ISI = INTER STIMULUS INTERVAL



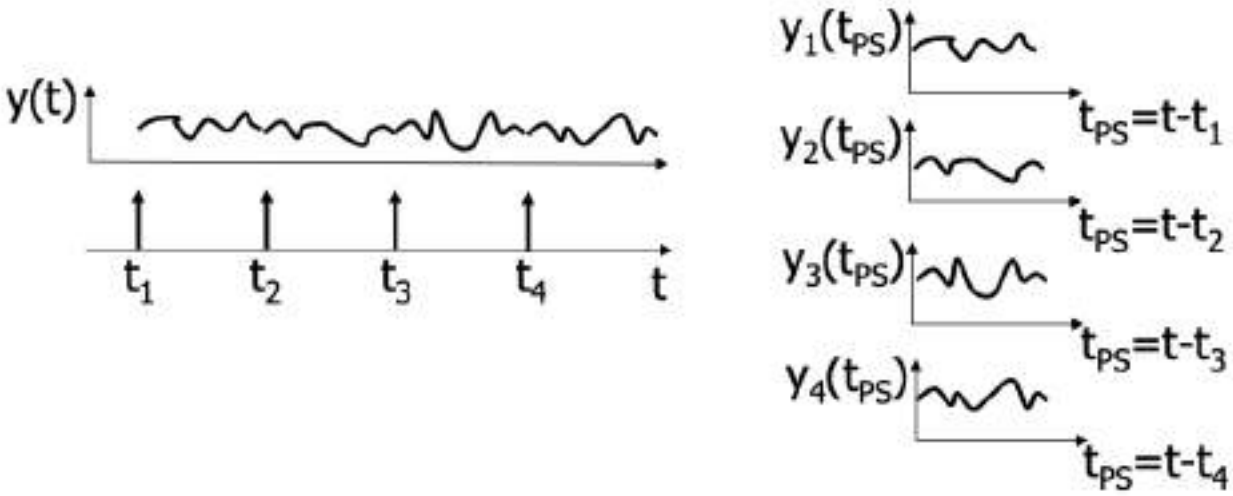
$y_1(t) = x_1(t) + v_1(t)$ $y_2(t) = x_2(t) + v_2(t)$ $y_N(t) = x_N(t) + v_N(t)$

Come si può rimuovere il rumore per identificare il PE ad una certa epoca?

Media sincrona

OPERATIVAMENTE

- Si ”taglia” il segnale in epoche
- Si mediano le epoche tra loro



Esercizio 1

I file DATA_Lab12_Esercizio1 contiene i potenziali evocati corticali misurati a seguito della stimolazione elettrica dell'esofago, con una frequenza di campionamento di 1000Hz. I segnali sono numerati come Ekk, dove k e' il numero della stimolazione, con k = 1, 2, . . . , M, e M = 5.

CONSEGNA

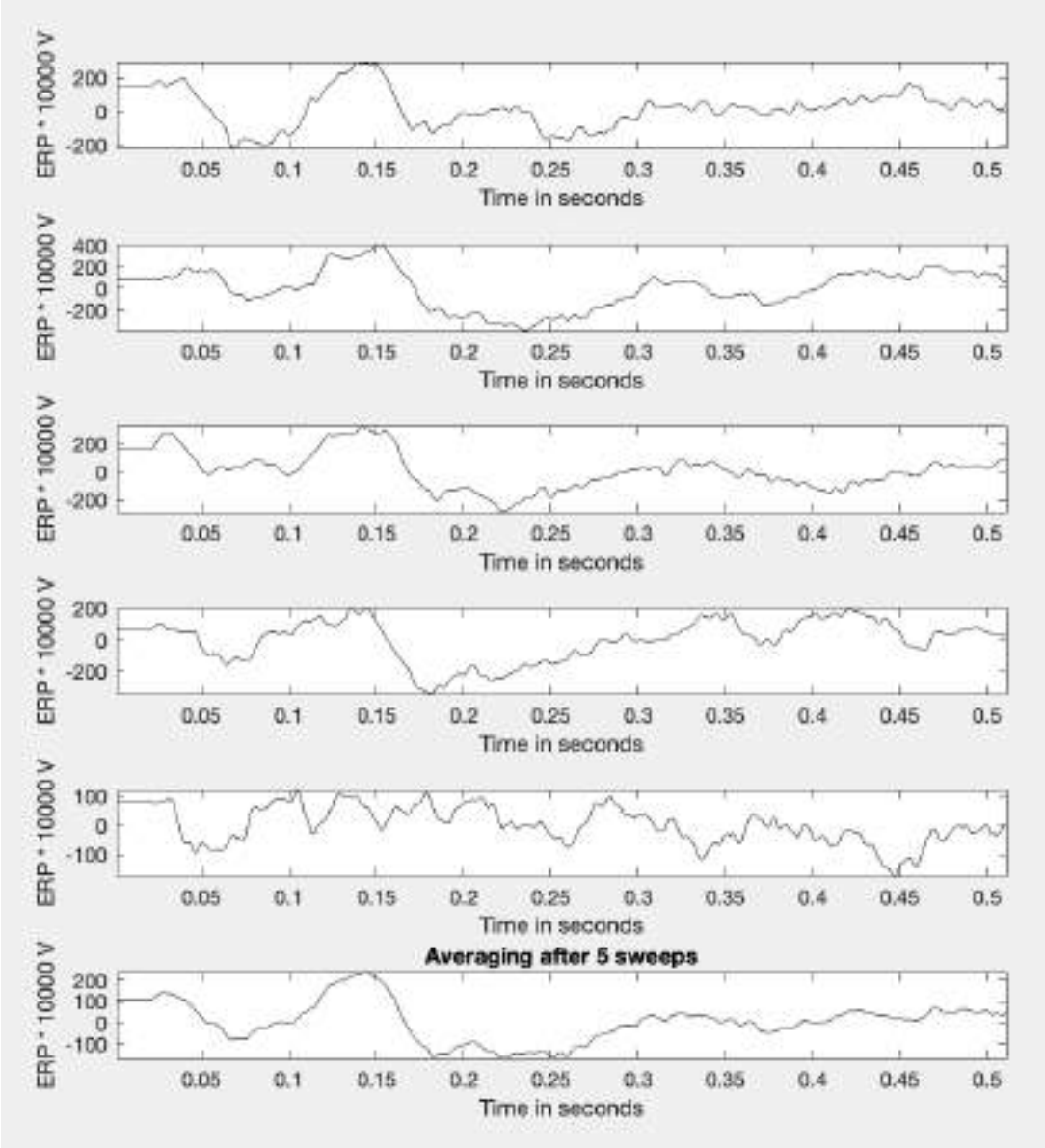
1) Carica e rappresenta graficamente i segnali Ekk in un'unica figura (utilizzando subplot)
2) Calcola e rappresenta graficamente la media sincrona dei segnali

Signal data courtesy of Dr. M. V. Kamath, McMaster University, Hamilton, ON, Canada.

```m
% Elaborazione Segnali Biomedici - Soluzione Laboratorio 12
% Prof. Veronese Mattia - UNIPD

%% OPENING
clear all
close all
clc


%% INIZIALIZZAZIONE DATI e VARIABILI

% Load cortical evoked potentials related to electrical stimulation of the esophagus
% Data courtesy of Dr. M.V. Kamath,
% McMaster University, Hamilton, ON, Canada.
load DATA_Lab12_Esercizio1

%Inizializzazione dati
d(:,1) = PE.E11;
d(:,2) = PE.E22;
d(:,3) = PE.E33;
d(:,4) = PE.E44;
d(:,5) = PE.E55;

% Parameters of the data acquisition
fs = 1000;      % sampling rate
T = 1/fs;     % sampling interval in seconds
N = size(d,1); % number of samples in each signal
tend = (N-1)*T;
time = linspace(0, tend, N);

%% Synchronous average (MEDIA SINCRONA)

% Compute the average of the trials
y = mean(d,2);

%% PLOT

% Plot multiple ERPs in the same figure
k_min = 1;              % first signal to be plotted
k_max = size(d,2);      % last signal to be plotted

figure(1);

for k = k_min:k_max+1

    % Extract the current signal from the matrix
    if k <= k_max

        % Signal of trial k
        y_k = d(:,k);

        % Plot the current signal in the proper subplot
        subplot(k_max+1, 1, k);
        plot(time, d(:,k), 'k-');
        title(['ERP #',num2str(k)]);
        xlabel('Time (secs)');
        ylabel('ERP*10000 (V)');
        axis tight;

    else

        % Plot the average of the trials in the proper subplot
        subplot(k_max+1, 1, k);
        plot(time, y, 'k-');
        title('Synchronous Average 5 sweeps')
        xlabel('Time (secs)');
        ylabel('ERP*10000 (V)');
        axis tight;

    end

end
```

Soluzioni esercizio 1

$$\hat{x}(k) = \frac{1}{N}\sum_{i=1}^{N} y_i(k)$$



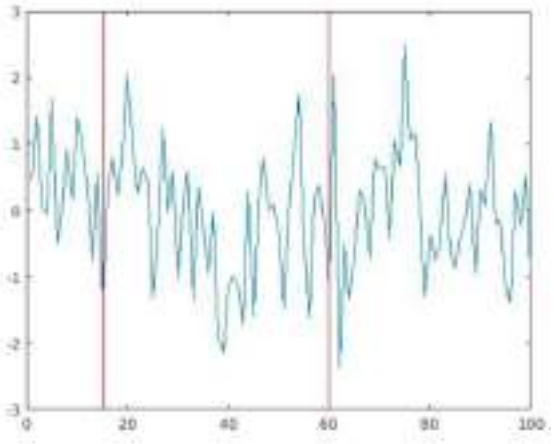
Esercizio 2

Il file DATA_Lab12_Esercizio2.mat contiene i dati EEG misurati durante un esperimento cognitivo. In questo esperimento, ogni 1300-1700 ms viene presentata una lettera o un numero. Il soggetto deve premere un bottone con la mano destra quando compare un numero e con la mano sinistra quando compare una lettera. Le lettere rappresentano circa il 20% del totale degli stimoli.

Nel file sono contenute le seguenti variabili:

- pz: contiene il dato EEG misurato sull'elettrodo Pz durante l'esecuzione compito. Tale segnale è stato campionato con una frequenza di 500 Hz.
- Time_Frequent_Digit e time_Rare_Letter: contiene gli istanti temporali in cui il soggetto vede rispettivamente un numero o una lettera.

Es.
X=randn(100,1);
plot(x)
hold on
plot([15 60; 15; 60], ylim,'r')



Esercizio 2

CONSEGNA:

1) Plottare il segnale EEG corrispondente alla finestra temporale 20 – 30 s, inserendo delle linee verticali di colore diverso in corrispondenza di uno stimolo frequente (comparsa di un numero) o di uno stimolo raro (comparsa di una lettera).

2) Calcolare il numero totale di stimoli (con numeri e lettere) ricevuti durante l'esperimento. Creare un vettore contenente tutti gli stimoli (numeri e lettere) in ordine temporale di comparsa.

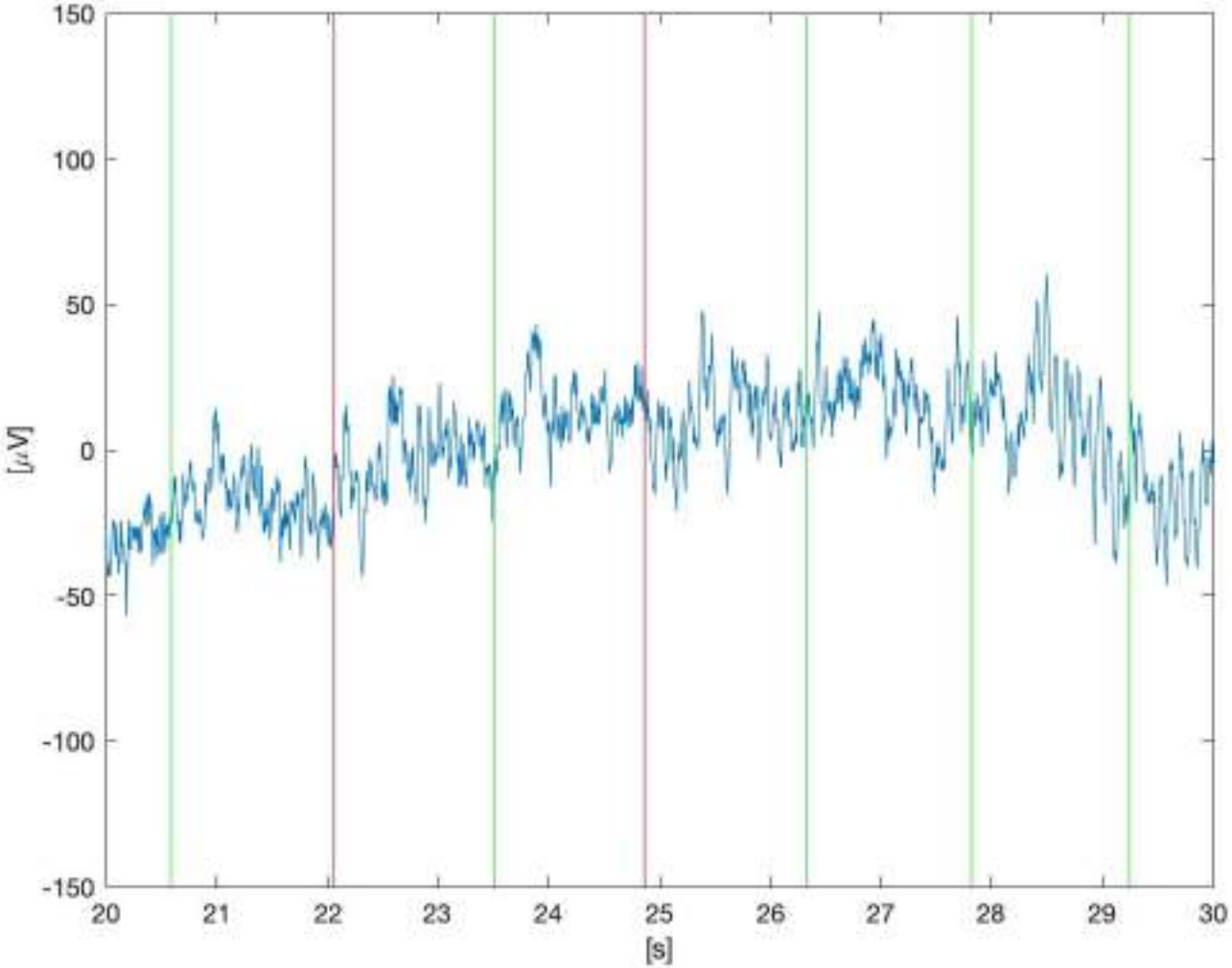
3) Dividere il segnale in segmenti corrispondenti all' attività registrata dal momento in cui compare lo stimolo fino ad 1 secondo dopo.

4) Dai segmenti ricavati nel punto 3), estrapolare quelli corrispondenti all'attività registrata dal momento in cui compare uno stimolo con lettera, calcolare la media sincrona e plottare il segnale risultante.

```
%% Esercizio 2
% Caricare i dati contenuti in DATA_Lab12_Esercizio2.mat
load DATA_Lab12_Esercizio2

%%
% 1) Visualizzare il segnale EEG corrispondente alla finestra temporale 20 – 30 s,
% inserendo delle linee verticali di colore diverso in corrispondenza della
% comparsa di un numero o di una lettera.

Fs   = 500; % Frequenza di campionamento [Hz]
time = [1/Fs:1/Fs:length(pz)/Fs];

figure(1),
plot(time,pz)
hold on,
plot([time_Frequent_Digit;time_Frequent_Digit],ylim,'g')
hold on,
plot([time_Rare_Letter;time_Rare_Letter],ylim,'r')
xlim([20 30])
xlabel('Time [s]')
ylabel('ERP [\muV]')
title('Segnale Misurato Pz')
legend('Signal','Digit Stimulus','Letter Stimulus')


%%
% 2) Calcolare il numero totale di stimoli (con numeri e lettere) ricevuti
% durante l'esperimento. Crerae un vettore contenente tutti gli stimoli in
% ordine di tempo di comparsa.

points_digit_raw    = int32(time_Frequent_Digit*Fs);        % Stimoli con numero (frequenti)
points_letter_raw   = int32(time_Rare_Letter*Fs);           % Stimoli con lettere (rari)
onsets      = sort([points_digit_raw points_letter_raw]);   % Totale stimoli (con numeri & lettere)
N_tot_onsets = length(onsets);                              % Numero totale di stimoli


%%
% 3) Dividere il segnale in segmenti corrispondenti all'attività registrata dal momento
% in cui compare lo stimolo fino ad 1 secondo dopo.

epoch_length    = 1;
points_epoch    = epoch_length*Fs;  % Totale campioni per ogni epoca
time_epoch        = [1/Fs:1/Fs:points_epoch/Fs];
epoch       = zeros(length(onsets),points_epoch);           % Totale epoche

for i = 1:length(onsets)
    current_epoch = pz(onsets(i):onsets(i)+points_epoch-1);
    epoch(i,:)    = current_epoch;
end


%%
% 4) Dai segmenti ricavati dal punto 2), estrapolare quelli corripondenti all'attivita' registrata
% dal momento in cui compare uno stimolo con lettera, calcolare la media
% mobile e plottarla

epoch_letter       = zeros(length(points_letter_raw),points_epoch);          % Totale epoche

for i = 1:length(points_letter_raw)
    current_epoch_letter = pz(points_letter_raw(i):points_letter_raw(i)+points_epoch-1);
    epoch_letter(i,:)    = current_epoch_letter;
end

% figure(2)
% plot(time_erp,epoch_letter)

% Media sincrona
mean_letter = sum(epoch_letter,1)/size(epoch_letter,1);

figure(2)
plot(time_epoch, mean_letter)
xlabel('Time [s]')
ylabel('ERP [\muV]')
title('Media sincrona')
```

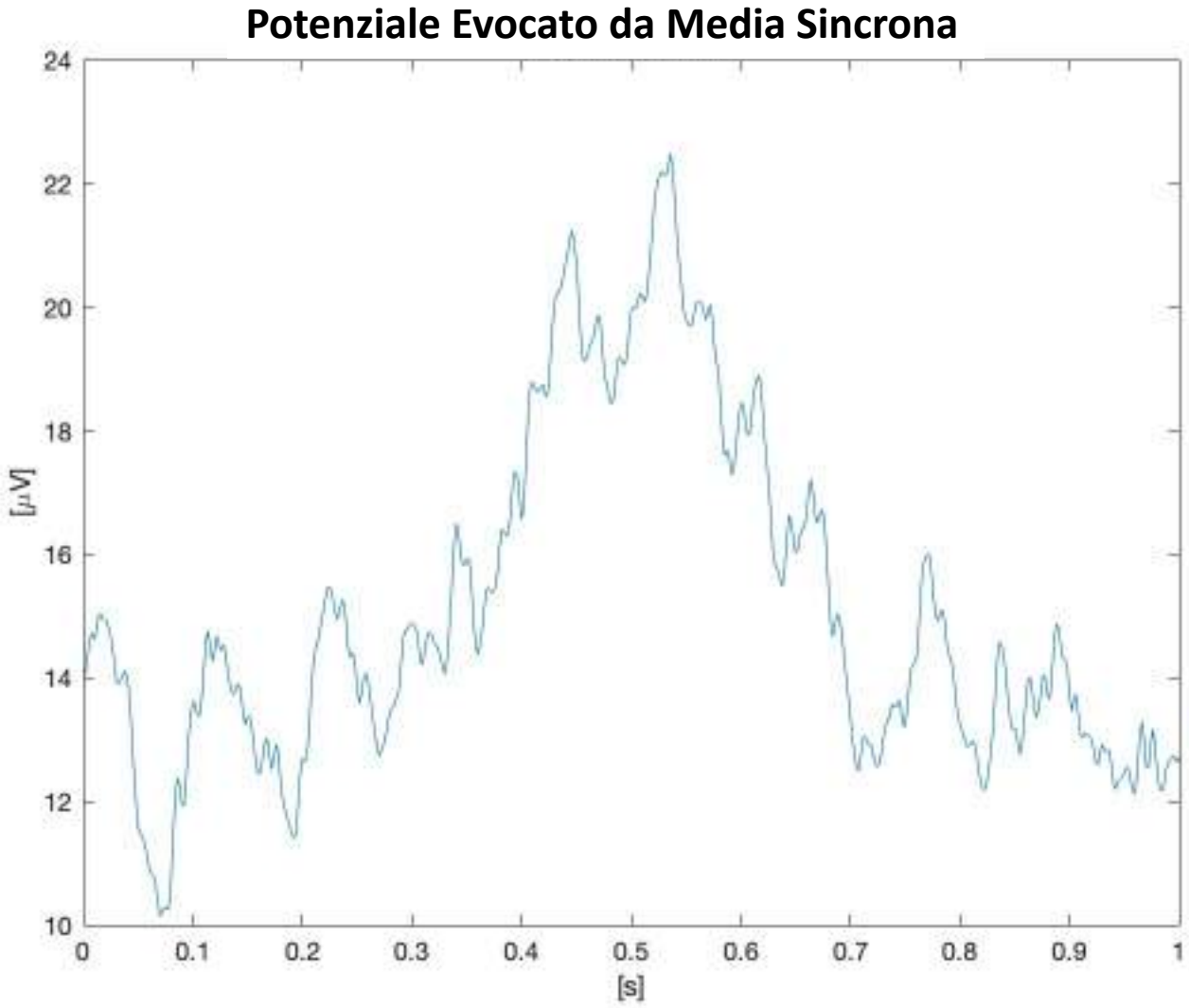
Soluzioni esercizio 2

1) **REGISTRAZIONE SEGNALE EEG NELLE VARIE EPOCHE**



Soluzioni esercizio 2

4)



Filtri nel dominio del tempo

Tutor: Dr. Giovanna Nordio e Giulia Vallini

Prof. Mattia Veronese
Email: mattia.veronese@unipd.it
Dipartimento di Ingegneria dell'Informazione

Ricevimento: su appuntamento (e-mail)
Edificio DEI/A, piano 2o, stanza 214

Sistemi LTI – Convoluzione

L'uscita del sistema è ricavabile tramite la convoluzione tra il segnale d'ingresso e la sequenza h(n) (= risposta del sistema all'impulso unitario)

$$y(n) = \sum_{k=-\infty}^{+\infty} x(k)h(n-k) = x * h$$

In **Matlab** si realizza usando la funzione conv:

y = conv(x,h)

dove x = vettore dei dati di ingresso e h = risposta impulsiva del sistema

Esercizio 1

Scrivere un M-file che rappresenti (comando *stem*) su tre riquadri (comando *subplot*) usando per tutti i segnali gli stessi assi di visualizzazione:

1) la sequenza h(n), con h(n)=1 per n=0,1,2,....,10 e h(n)=0 altrove
2) la sequenza x(n), dove x(n) è l'esponenziale di base 0.5 con n ∈ [0, 10]
3) la loro convoluzione calcolata utilizzando la funzione **conv**

```
% Elaborazione Segnali Biomedici - Soluzione Laboratorio 14
% Prof. Veronese Mattia - UNIPD

%% OPENING
clear all
close all
clc

%% Esercizio 1
% Calcolare l'uscita del sistema avente h(n)=1 n=0,1,2,.....,10 e h(n)=0 altrove
% in risposta all'ingresso x(n)dove x è l'esponenziale di base 0.5 in (0,10)
% Plottare h, x e y usando il comando stem

n = [0:1:10];
h = ones(size(n));
x = (0.5).^n;

% Uscita del sistema
y = conv(h,x);
y = filter(h,1,x);

% Rappresentazione grafica dell'uscita del sistema h, del segnale in
% ingresso x e del segnale in uscita y
figure
subplot(3,1,1)
stem(n,h)
xlabel('n')
ylabel('h(n) (u.a.)')
title('h')
subplot(3,1,2)
stem(n,x)
xlabel('n')
ylabel('x(n) (u.a.)')
title('ingresso')
subplot(3,1,3)
stem(n,y(1:length(n))) %È necessario per la conv ridurre l'uscita nell'intervallo dei segnali di input
xlabel('n')
ylabel('y(n) (u.a.)')
title('uscita')
```

Esercizio 2

Scrivere un M-file che rappresenti (comando stem) su tre riquadri (comando subplot) usando per tutti i segnali gli stessi assi di visualizzazione (comando axis([0 25 0 3])):

1) la sequenza h(n), dove h(n) è l'esponenziale di base 0.5 con $n \in [0, 10]$
2) la sequenza x(n), dove x(n) vale 1.2 con $n \in [0, 10]$
3) la loro convoluzione calcolata utilizzando la funzione **conv**

```m
% Elaborazione Segnali Biomedici - Soluzione Laboratorio 14
% Prof. Veronese Mattia - UNIPD

%% OPENING
clear all
close all
clc


%% Esercizio 2

n = [0:1:10];
% Risposta h(n)
h = (0.5).^n;
% Ingresso x(n)
x = 1.2*ones(size(n));
% Uscita
y = conv(h,x);

% Rappresentazione del segnale h, x e y
figure
subplot(3,1,1)
stem(n,h)
xlabel('n')
ylabel('h(n) (u.a.)')
title('Risposta Impulsiva')
axis([0 25 0 3])
subplot(3,1,2)
stem(n,x)
xlabel('n')
ylabel('x(n) (u.a.)')
title('INGRESSO')
axis([0 25 0 3])
subplot(3,1,3)
stem(n,y(1:length(n)))
xlabel('n')
ylabel('y(n) (u.a.)')
title('USCITA')
axis([0 25 0 3])


```

Sistemi LTI – Equazioni alle differenze

L'uscita del sistema è ricavabile tramite la convoluzione tra il segnale d'ingresso e la sequenza h(n) (= risposta del sistema all'impulso unitario)

$$y(n) = -\sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k\, x(n-k)$$

In **Matlab** si realizza usando la funzione filter :

y = filter(b,a,x)

dove x = vettore dei dati di ingresso
b=[b_0,b_1 … b_M] coefficienti della parte MA
a=[1, a_1 … a_N] coefficienti della parte AR

Esercizio 3

Calcolare l'uscita del sistema

$$y(n) = x(n)+ x(n-1) + x(n-2) + x(n-3) + x(n-4)$$

in risposta all'ingresso x, l'esponenziale di base 0.5, nell'intervallo di tempo [0,10] secondi con tempo di campionamento Tc=0.1 secondi.

Plottare i segnali ingresso x e uscita y rispetto al tempo.

```m
% Elaborazione Segnali Biomedici - Soluzione Laboratorio 14
% Prof. Veronese Mattia - UNIPD

%% OPENING
clear all
close all
clc

%% Esercizio 3
% Calcolare l'uscita del sistema
% y(n) = x(n)+ x(n-1) + x(n-2) + x(n-3) + x(n-4)
% in risposta all ingresso x, l'esponenziale di base 0.5 definito
% nell'intervallo di tempo (0,10) con tc=0.1 secondi
A = [1];
B = [1 1 1 1 1];
tc = 0.1; %secondi
time = [0:tc:10]; %secondi
x = (0.5).^time;  %ingresso


% Uscita del sistema
y = filter(B,A,x);


% Plottare i segnali ingresso x e uscita y rispetto al tempo
figure
subplot(2,1,1)
stem(time,x)
xlabel('Time [secondi]')
ylabel('x(t)')
title('Ingresso')
subplot(2,1,2)
stem(time,y)
xlabel('Time [secondi]')
ylabel('y(t)')
title('Uscita')
```

Esercizio 4

Si consideri come ingresso al sistema

$$y(n) = x(n)+ x(n-1) + x(n-2) + x(n-3) + x(n-4)$$

il seguente segnale

$$x(n)=\sin(2\pi f0\ nTs)$$

che rappresenta una sinusoide a frequenza f0 (Hz) campionata con periodo Ts=0.01 secondi.

- Calcolare ed osservare l'uscita in n=0,1, …, 30, confrontandola con l'ingresso, quando f0=20, f0=40, f0=30.
- Verificare che, nei primi due casi, l'uscita dopo alcuni passi diventa identicamente pari a zero, nonostante l'ingresso sia non nullo.

```m
% Elaborazione Segnali Biomedici - Soluzione Laboratorio 14
% Prof. Veronese Mattia - UNIPD

%% OPENING
clear all
close all
clc


%% Esercizio 4
% Realizzazione sistema MA

% Coefficienti del sistema
B = [1 1 1 1 1];
A = [1];

% Segnale id ingresso
n = [0:1:30]; %Campioni a disposizione
Ts = 0.01;    %Passo di campionamento
f0 = [20, 30, 40];


for i=1:length(f0)

    x = sin(2*pi*f0(i)*n*Ts);
    y = filter(B,A,x);

    % Rappresentazione grafica dei segnali
    figure(i)
    subplot(2,1,1)
    stem(n,x)
    ylabel('x(n) (u.a.)')
    title(['ingresso (f0=' num2str(f0(i)) ')'])
    subplot(2,1,2)
    stem(n,y)
    xlabel('n (u.a.)')
    ylabel('y(n) (u.a.)')
    title('uscita')

end

```

Filtri nel dominio della frequenza

Tutor: Dr. Giulia Vallini

Prof. Mattia Veronese
Email: mattia.veronese@unipd.it
Dipartimento di Ingegneria dell'Informazione

Progettazione filtri digitali

La progettazione di un filtro si può fare:

- Tramite **allocazione zeri e poli** della funzione di trasferimento H(z) nel piano complesso z, verificando sempre STABILITA' (poli nel cerchio di raggio unitario) e REALIZZABILITA' FISICA (causalita' e coefficienti reali)
- Tramite metodi di sintesi al calcolatore con funzioni build- in di MATLAB

Progettazione filtri digitali

REALIZZABILITÀ FISICA

Causalità della sequenza *h(n)* *IMPLICA che*

- la regione di convergenza di *H(z)* corrisponde all'esterno di un cerchio di raggio > del polo di *H(z)* di valore assoluto massimo
- Il numero di poli è sempre maggiore o uguale al numero di zeri

Coefficienti reali di *h(n)* *IMPLICA che*

- per ogni polo e zero complesso, deve essere presente anche il rispettivo complesso coniugato
- poli e zeri sull'asse reale possono essere singoli (hanno parte immaginaria nulla quindi coincidono con i propri complessi coniugati).

STABILITÀ

- tutti i polidel filtro devono cadere nel cerchio unitario, mentre gli zeri possono essere posizionati in qualunque punto del piano z

Calcolo di H(z) a partire dai suoi poli e zeri

Per calcolare H(z):

1) si definiscono i vettori **z** vettore degli zeri e **p** vettore dei poli

2) tramite la funzione matlab **poly** si ricavano b ed a (coefficienti del numeratore e denominatore di H(z))

b=poly(z)
a=poly(p)

3) si calcola H(z) con la funzione matlab freqz

[H,F] = freqz(b,a,Np,Fc)

N.B. Se è nota H(z) si hanno il vettore dei coefficienti del numeratore (b) e quello dei coefficienti del denominatore (a). Si possono quindi calcolare zeri e poli tramite la funzione **roots**:

z=roots(b) → zeri di H
p=roots(a) → poli di H (N.B. per stabilità abs(p)< 1)

Calcolo di H(z) a partire dai suoi poli e zeri

Per disegnare il diagramma di Bode:

freqz(b,a,Np,Fc)

Oppure:

```
[H,F] = freqz(b,a,2048,Fs);
figure;
subplot(2,1,1)
plot(F,20*log10(abs(H)))      % Disegna il modulo
title('modulo'); xlabel('Hz')
subplot(2,1,2)
plot(F,angle(H)*360/(2*pi))   % Disegna la fase
title('fase'); xlabel('Hz')
```

Per disegnare zeri e poli:

zplane(z,p) con z e p vettori colonna
zplane(b,a) con b e a vettori riga

Progettazione filtro: guadagno

Nella progettazione di un filtro, una volta stabiliti i poli (sempre dentro al cerchio unitario) e gli zeri, si deve imporre che il guadagno sia unitario per le frequenze da far passare inalterate.

Per fare cio' si usa la funzione:

polyval(b,x) *valuta il polinomio b in x*

Per garantire che la frequenza resti inalterata, si calcola H(z) nel punto z_0, corrispondente appunto a ω_0, e si pone:

```
G= polyval(b,z0)/polyval(a,z0);      % Valore di H(z) in z0
b= b*(1/G);
[H, F] = freqz(b,a,2048,Fc); → |H(z0)|=1
```

Esercizio 1

Progettare un filtro con uno zero in -1, un polo in 0.9, una coppia di zeri e poli complessi coniugati a 30 Hz e un'altra a 60 Hz. Gli zeri siano sul cerchio di raggio unitario e i poli all'interno del cerchio unitario (ad es. modulo 0.9).
Imporre che il guadagno sia unitario per z_0=1.

QUESITI:

- Visualizzare la risposta in frequenza del filtro, in modulo e fase, e il diagramma zeri-poli.

- Applicare tale filtro al segnale campionato a 1000 Hz contenuto nel file segnale_bioelettrico.mat, e plottare il segnale originale e quello filtrato.

```m
% Elaborazione Segnali Biomedici - Soluzione Laboratorio 22
% Prof. Veronese Mattia - UNIPD

%% OPENING
clear all
close all
clc

%% Esercizio 1 - CHALLENGE
% Progettare un filtro con uno zero in -1, un polo in 0.9, una coppia di
% zeri e poli complessi coniugati a 30 Hz e un'altra a 60 Hz.
% Gli zeri siano sul cerchio di raggio unitario e i poli all'interno del
% cerchio unitario (ad es. modulo 0.9).

Fs = 1000;
F1 = 60;
theta1 = (2*pi/Fs)*F1;
F2 = 30;
theta2 = (2*pi/Fs)*F2;

zeri=[exp(1i*theta1), exp(-1i*theta1) exp(1i*theta2), exp(-1i*theta2) -1 ];
coeff=0.9;
poli=[coeff*exp(1i*theta1), coeff*exp(-1i*theta1), coeff*exp(1i*theta2), coeff*exp(-1i*theta2) coeff];

b=poly(zeri);
a=poly(poli);

G=polyval(b,1)/polyval(a,1);
b=b/G;
[H,F]=freqz(b,a,2048,Fs);

% Diagramma di Bode del filtro
figure
subplot(2,1,1)
plot(F,20*log10(abs(H)))
title('modulo')
xlabel('Hz')
subplot(2,1,2)
plot(F,angle(H)*360/(2*pi))
title('fase')
xlabel('Hz')

% Diagramma zeri-poli
figure
zplane(b,a)


%% FILTRARE IL SEGNALE

% Load ECG data
load segnale_bioelettrico.mat

dati = segnale;
N = length(dati);
t = [0:1/Fs:(N-1)/Fs];
y = filter(b,a,dati);

figure
plot(t, dati, 'b')
hold on
plot(t, y, 'r')
xlabel('time')
ylabel('u.a.')
legend('Segnale originale', 'Segnale filtrato')

```

Esercizio 2

Progettare un filtro in modo che la funzione di trasferimento H(z) del segnale campionato con frequenza di 1000 Hz abbia:

1. Uno zero in z=0 e un polo in p=0.8
2. Uno zero in 0.5 e una coppia di poli complessi coniugati in -0.6±j0.3
3. Uno zero in 0.5 e una coppia di poli complessi coniugati a 60Hz dentro al cerchio unitario

MATLAB Drive > Lab19_Es2_9f26513d7ba880dc3537c494629e5a89.m

```
% Elaborazione Segnali Biomedici - Soluzione Laboratorio 22
% Prof. Veronese Mattia - UNIPD

%% OPENING
clear all
close all
clc

%% Esercizio 2
% Progettare un filtro in modo che la funzione di trasferimento H(z) abbia:
% Uno zero in z=0 e un polo in p=0.8
Fc = 1000;
zeri = 0;
poli = 0.8;
b = poly(zeri);
a = poly(poli);

figure
zplane(b,a)
[H,F]=freqz(b,a,2048,Fc);


% Uno zero in 0.5 e una coppia di poli complessi coniugati in -0.6±j0.3
Fc = 1000;
zeri = 0.5;
poli = [-0.6+0.3j -0.6-0.3j];
b = poly(zeri);
a = poly(poli);

figure
zplane(b,a)
[H,F]=freqz(b,a,2048,Fc);


% Uno zero in 0.5 e una coppia di poli complessi coniugati a 60Hz dentro al
% cerchio unitario
Fc = 1000;
F = 60;
theta=(2*pi/Fc)*F;
zeri = 0.5;
poli = [0.9*exp(1i*theta), 0.9*exp(-1i*theta)];
b = poly(zeri);
a = poly(poli);

figure
zplane(b,a)
[H,F]=freqz(b,a,2048,Fc);
```

Esercizio 3

Si consideri il filtro dato dalla seguente equazione alle differenze, frequenza di campionamento 1000 Hz:

$$y(n) + 0.81y(n-1) = x(n) - x(n-2)$$

1. Determinare la funzione di trasferimento del filtro
2. Determinare poli e zeri della funzione di trasferimento
3. Visualizzare la risposta in frequenza del filtro, in modulo e fase, e il diagramma zeri-poli

MATLAB Drive > Lab19_Es3_0d8cdc14c57ea7b90165375483a81996.m

```
% Elaborazione Segnali Biomedici - Soluzione Laboratorio 22
% Prof. Veronese Mattia - UNIPD

%% OPENING
clear all
close all
clc

%% Esercizio 3

% 2) Determinare poli e zeri della funzione di trasferimento e farne il
% grafico nel piano di Gauss
% H(z) = (1-z^(-2))/(1+0.81z^(-1))

% Coefficienti del numeratore e denominatore
a = [1 0.81];
b = [1 0 -1];

% Zeri e poli
z = roots(b);
p = roots(a);

%%
% 3) Visualizzare la risposta in frequenza del filtro, in modulo e fase, e il
% diagramma zeri-poli
Fc = 1000;
Np = 2048;

[H, F] = freqz(b,a,Np,Fc);

figure
subplot(2,1,1)
plot(F,20*log10(abs(H)))    % Disegna il modulo
title('modulo')
xlabel('Hz')
subplot(2,1,2)
plot(F,angle(H)*360/(2*pi))     % Disegna la fase
title('fase')
xlabel('Hz')

% Diagramma zeri-poli
figure
zplane(b,a)

```

Esercizio 4

Progettare un filtro passa-basso con 2 poli reali coincidenti in 0.8 e 2 zeri nell'origine, frequenza di campionamento di 1000 Hz, in modo che abbia guadagno unitario per ω=0 [$|H(\omega=0)| = |H(1)| = 1$].

Visualizzare la risposta in frequenza del filtro, in modulo e fase, e il diagramma zeri-poli.

MATLAB Drive > Lab19_Es4_6b9045290d7be546b17f53e9b842ca3a.m

```
% Elaborazione Segnali Biomedici - Soluzione Laboratorio 22
% Prof. Veronese Mattia - UNIPD

%% OPENING
clear all
close all
clc

%% Esercizio 4
% Progettare un filtro passa-basso con 2 poli reali coincidenti in 0.8 e
% 2 zeri nell' origine, in modo che abbia guadagno unitario per ω=0
% |H(ω=0)|= |H(1)|= 1
% Visualizzare la risposta in frequenza del filtro, in modulo e fase,
% e il diagramma zeri-poli.


Fc = 1000;
poli = [0.8, 0.8];
zeri = [0, 0];
b = poly(zeri);
a = poly(poli);

G = polyval(b,1)/polyval(a,1);
b = b/G;

% Filtro
[H, F] = freqz(b,a,2048,Fc);

figure
subplot(2,1,1)
plot(F,20*log10(abs(H)))  % Disegna il modulo
title('modulo')
xlabel('Hz')
subplot(2,1,2)
plot(F,angle(H)*360/(2*pi))     % Disegna la fase
title('fase')
xlabel('Hz')

% Diagramma zeri-poli
figure
zplane(b,a)
```

Analisi spettrale

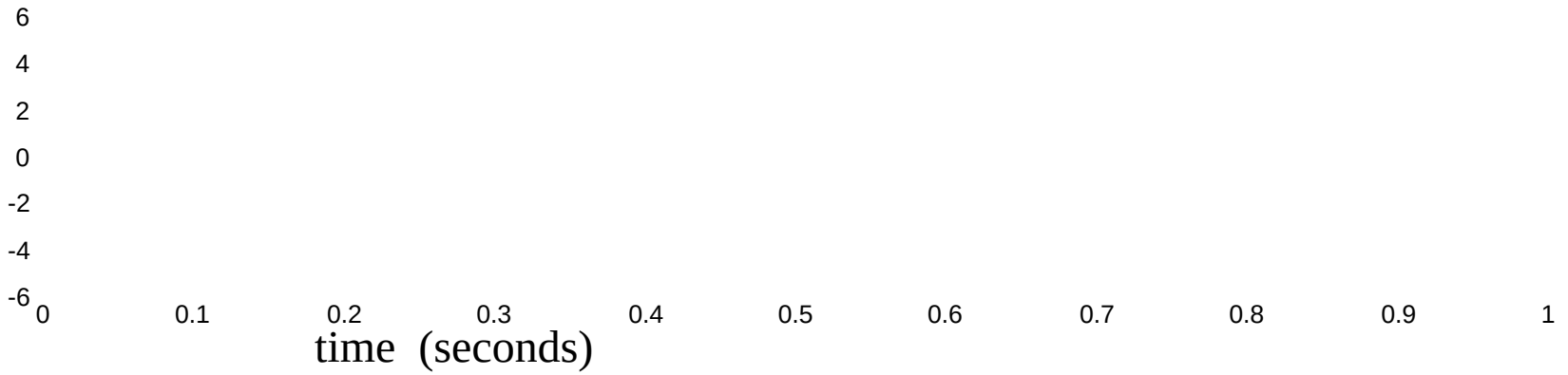
Tutor: Dr. Giovanna Nordio e Giulia Vallini

Prof. Mattia Veronese
Email: mattia.veronese@unipd.it
Dipartimento di Ingegneria dell'Informazione

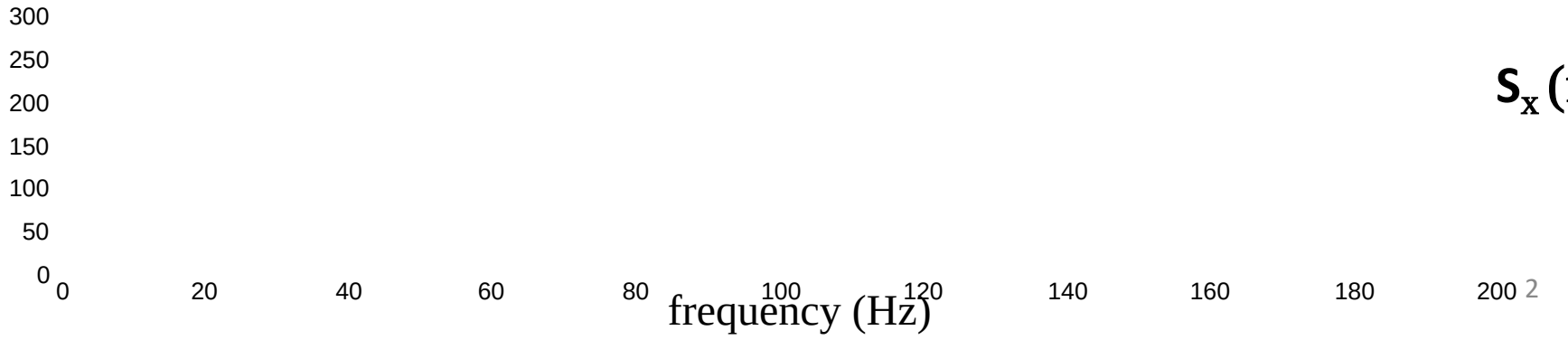
Ricevimento: su appuntamento (e-mail)
Edificio DEI/A, piano 1o, stanza 106

Analisi spettrale

Esempio: Si consideri il segnale sotto, riferito ad una frequenza di sampling Fs = 400 Hz ed osservato per 1 secondo



Dal grafico (dominio del tempo) non si sa che natura attribuire a x(t), mentre dal suo spettro $S_x(f)$ (dominio della frequenza), si vede che x(t) non è altro che la somma di due sinusoidi a 50 e 120 Hz immerse in rumore bianco



Analisi spettrale

Come stimare lo spettro di un segnale?

- **Metrodi FT-based:**
  - Metodo diretto o periodogramma
  - Metodo indiretto

- **Metodi parametrici**

Metodo diretto o periodogramma

E' basato sulla definizione di **Densità Spettrale di Potenza** di un segnale discreto x(n), n=0...N-1, campionato con frequenza Fs, per cui:

$$P(\omega) = \frac{1}{2}|X(\omega)|^2 \qquad \text{dove } X(\omega)=FT[x(n)]$$

Nella realta' si calcola una DFT, con l'istruzione Matlab **X=fft(x,N)**, e quindi:

P=(abs(X).^2)/N

X è un vettore complesso di N campioni, corrispondenti ad N frequenze equispaziate tra 0 e Fs → **f_FT=(0:Fs/N:Fs-Fs/N)**

abs(X) → vettore modulo
angle(X) → vettore della fase

Metodo diretto o periodogramma

Per riassumere, i comandi MATLAB da usare per calcolare la densita' spettrale di potenza sono:

```
FTx=fft(x,N)

S=(abs(FTx).^2)/N

f_FT=(0:Fs/N:Fs-Fs/N)
```

Ma per rappresentare lo spettro in figura, basta visualizzare da 0 a Fs/2

```
plot(f_FT(1:N/2),S(1:N/2))
```

Metodo diretto o periodogramma

Questo metodo presuppone che:

- non ci sia aliasing
- si considerino solo N/2 punti relativi alle frequenze tra 0 e Fs/2

Problemi:

- Se il segnale è stato troncato c'è errore di distorsione (leakage) ➔ ***FINESTRATURA*** (il calcolo migliora quanto maggiore è la finestra di osservazione)
- Se N è piccolo si calcola lo spettro solo per poche frequenze ➔ ***ZERO-PADDING***

In MATLAB e' possibile nella DFT introdurre zero-padding con l'istruzione:

DFTx=fft(x, Nzp)

con Nzp>N, dove N e' la durata di x

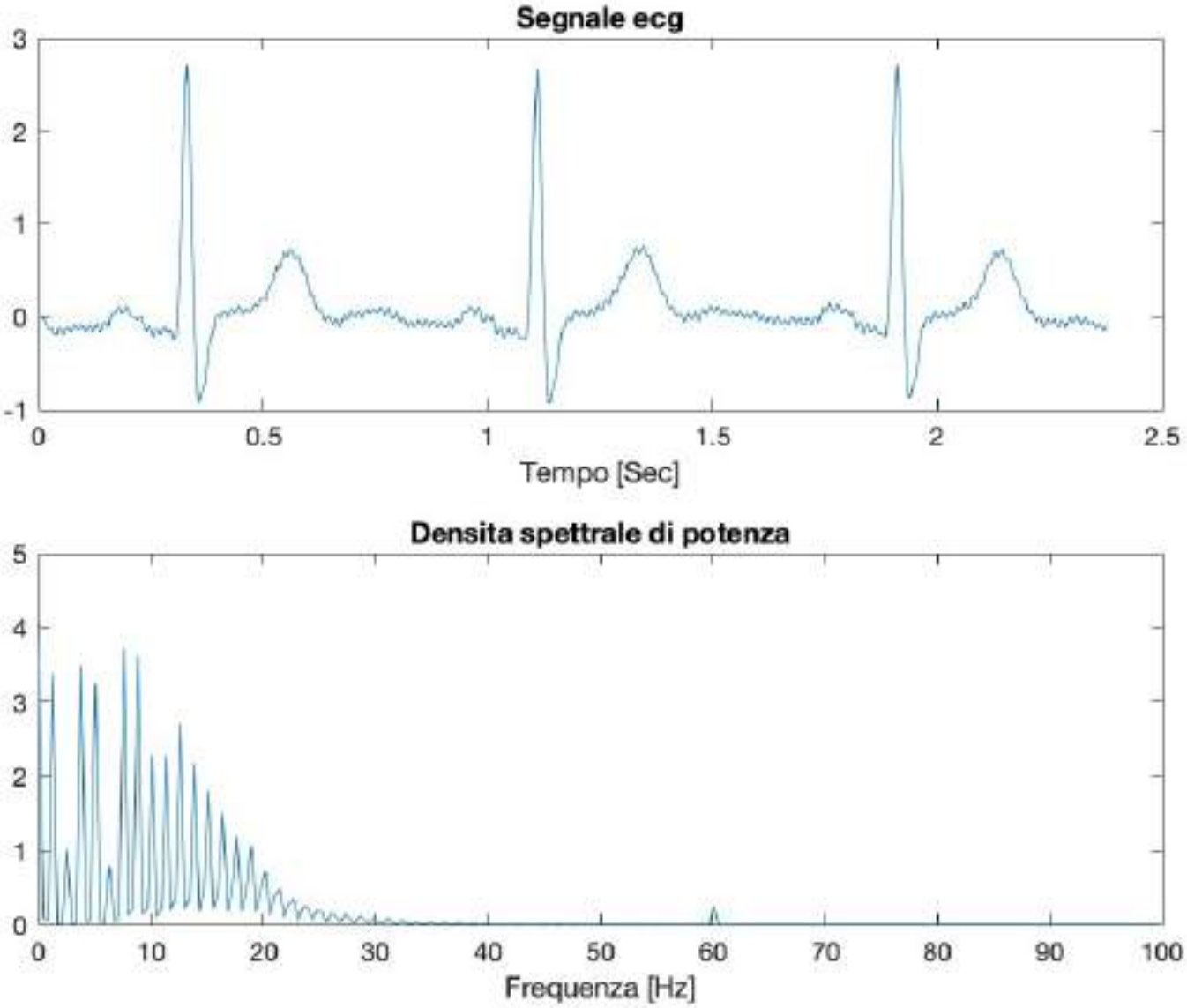
Esercizio 1

Calcolare lo spettro del segnale contenuto nel file ecg_60.mat, campionato a 200Hz.
Calcolare la densita' spettrale di potenza media nell'intervallo [0,50]Hz e [50,100]Hz.
In quale range di frequenze si concentra la maggior parte della potenza del segnale?

MATLAB Drive > Lab24_Es1_c1b1719d55451f0310dcde334098bd9e.m

```
% Elaborazione Segnali Biomedici - Soluzione Laboratorio 24
% Prof. Veronese Mattia - UNIPD

%% OPENING
clear all
close all
clc

%% Esercizio 1
% Calcolare lo spettro del segnale ecg contenuto nel file ecg_60.mat,
% campionato a 200Hz.

load ecg_60.mat

N = length(t);  % Totale campioni
Fs = 200;       % Frequenza di campionamento
x = ecg_60;     % segnale

figure(1)
subplot(2,1,1)
plot(t,x)
xlabel('Tempo [Sec]')
title('Segnale ecg')

% Calcolo spettro del segnale
FTx = fft(x,N);
S = (abs(FTx).^2)/N;
f_FT = (0:Fs/N:Fs-Fs/N);
subplot(2,1,2)
plot(f_FT(1:N/2),S(1:N/2))
xlabel('Frequenza [Hz]')
title('Densita spettrale di potenza')

% Densita' spettrale di potenza media nell'intervallo [0,50] e [50,100] Hz
% Intervallo [0,50]Hz
f_start_first_interval = 0;
f_end_first_interval = 50;
ind = intersect(find(f_FT>=f_start_first_interval),find(f_FT<=f_end_first_interval));
mean_power_first_interval = mean(S(ind));

% Intervallo [50,100]Hz
f_start_second_interval = 50;
f_end_second_interval = 100;
ind = intersect(find(f_FT>=f_start_second_interval),find(f_FT<=f_end_second_interval));
mean_power_second_interval = mean(S(ind));


```

Soluzioni – Esercizio 1



Esercizio 2

Considerare un segnale sinusoidale a tempo continuo $x(t) = \sin(2\pi f_0 t)$, dove la frequenza e' $f_0=1/16$Hz , corrispondente ad un periodo di $T_0=16$s.

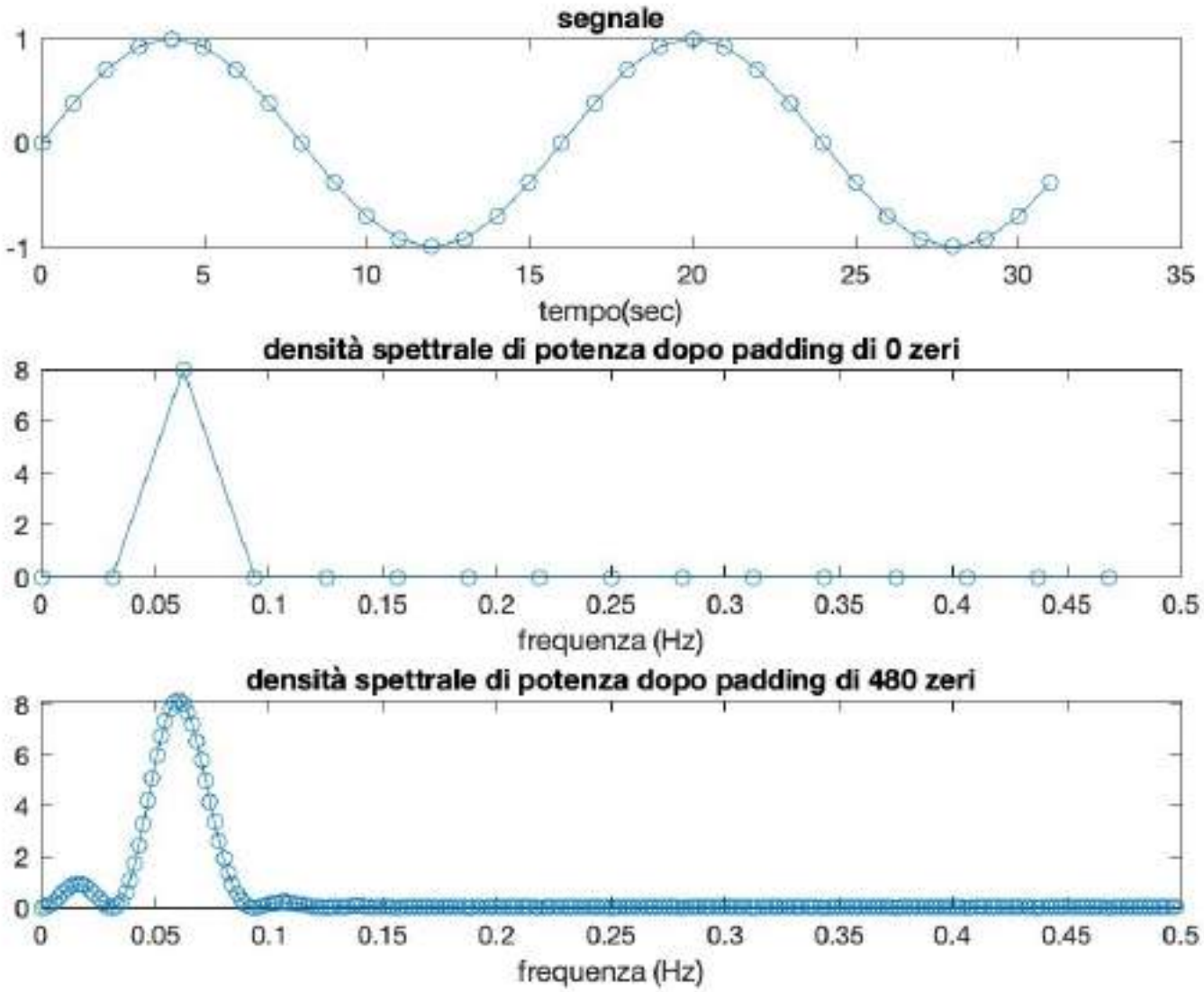
Considerare la sequenza di campioni $x(nT_s)$, n =0,1,...N-1, riferita ad un periodo di sampling $T_s=1$. Per ogni periodo di ripetizione del segnale x(t) sono quindi raccolti 16 campioni.

Si fissi N=32, corrispondente a 2 periodi completi di campionamento. Stimare lo spettro tra 0 e Fs/2.

Stimare poi lo spettro con Nzp=512 e plottarlo tra 0 e Fs/2. Confrontare lo spettro con quanto ottenuto senza zero-padding.

```
Fs=1;        % Frequenza di campionamento
Ts=1/Fs;     % Periodo di campionamento
T0=16;       % Periodo segnale
np=2;        % Periodi completi di campionamento

N=np*T0;     % Numero campioni
Nzp=512;     % Zero-padding


% Segnale sinusoidale
t=Ts*(0:1:N-1)';
x=sin(2*pi*t/T0);

subplot(3,1,1)
plot(t,x, 'o-')
xlabel('tempo(sec)')
title('segnale')


%% Spettro del segnale dopo padding di 0 zeri
FTx=fft(x);
S=(abs(FTx).^2)/N;
f_FT=(0:Fs/N:Fs-Fs/N);

f_FT=f_FT(1:N/2);    % Elimino la seconda meta' delle stime
S=S(1:N/2);          % Elimino la seconda meta' delle stime

subplot(3,1,2)
plot(f_FT,S,'o-')
axis([0, Fs/2, 0, max(abs(S))])
title('densità spettrale di potenza dopo padding di 0 zeri')
xlabel('frequenza (Hz)')


%% Spettro del segnale con zero-padding
FTx_zp=fft(x,Nzp);
S_zp=(abs(FTx_zp).^2)/N;
f_FT_zp=(0:Fs/Nzp:Fs-Fs/Nzp);

f_FT_zp=f_FT_zp(1:Nzp/2);    % Elimino la seconda meta' delle stime
S_zp=S_zp(1:Nzp/2);          % Elimino la seconda meta' delle stime

subplot(3,1,3)
plot(f_FT_zp,S_zp,'-o')
axis([0, Fs/2, 0, max(abs(S_zp))])
title('densità spettrale di potenza dopo padding di 480 zeri')
xlabel('frequenza (Hz)')
```

Soluzioni – Esercizio 2



ELABORAZIONE SEGNALI BIOMEDICI – A.A. 2022/2023
Prof. Veronese Mattia – Dr. Giovanna Nordio – Dr. Giulia Vallini

SIMULAZIONE ESAME - ESERCIZIO MATLAB

Tempo Max previsto per soluzione esercizio (inclusa consegna: 1h)

Dato il segnale **ECG.mat** che descrive l'acquisizione di un elettrocardiogramma di un soggetto sano (Frequenza di campionamento = 1000Hz), costruire uno script in grado di eseguire le seguenti operazioni:

1) Elaborare il segnale con un filtro a media mobile $y(n) = \frac{x(n)+x(n-1)+x(n-2)+x(n-3)}{4}$

2) Estrarre i tempi di picco associati ai vari complessi R del segnale filtrato

3) A partire dal tracciato dei picchi ottenuto nel punto 2), costruire il tacogramma del segnale ECG filtrato, andando a calcolare le durate dei singoli cicli cardiaci come distanza di due successivi picchi R-R

4) Dal tacogramma costruito nel punto 3), ottenere la misura del battito cardiaco medio e il suo range minimo e massimo

ISTRUZIONI PER LA CONSEGNA

Nella pagina moodle del corso, consegnare due file con le seguenti informazioni:

- File **ECG.fig** che confronti il segnale ECG originale con il segnale filtrato, e indichi i picchi R dei vari cicli cardiaci. La figura deve essere auto esplicativa. Non dimenticare titoli degli assi, unità di misura e legenda
- File **Risultati.mat** contente le seguenti variabili
    - **Durata**, variabile numerica contenente la durata in secondi dell'acquisizione ECG
    - **ECG_filtrato**, vettore numerico contenente il segnale filtrato ottenuto dall'applicazione del filtro a media mobile (Q1)
    - **Tacogramma**, vettore numerico contenente il tacogramma derivato dal segnale ECG_filtrato (Q3)
    - **N**, variabile numerica contenente il numero di cicli cardiaci identificati nel segnale ECG
    - **Battito_medio**, variabile numerica per il battito cardiaco medio espresso in bpm
    - **Battito_min**, variabile numerica per il battito cardiaco min espresso in bpm
    - **Battito_max**, variabile numerica per il battito cardiaco max espresso in bpm

RACCOMANDAZIONI
Attenzione a nominare file e variabili nel modo corretto (Matlab è case-sensitive).
La consegna dello script (file.m) è opzionale.

MATLAB Drive > esempi_esame.m

```
clear all
close all
clc

%% ESEMPIO TEMA D'ESAME
load ECG.mat
Fc = 1000; % [Hz]
T = 1/Fc; % [s]
a = [4];
b = [1 1 1 1];
time = 0:T:(length(ECG)-1)/Fc;
time = time';
Durata = time(end);
ECG_filtrato = filter(b,a,ECG);
tempi_picchi = [0.292,1.008,1.712,2.434,3.159,3.849,4.571,5.322,6.011,6.726,7.451,8.166];
picchi = [2.60334,2.64008,2.63515,2.571,2.5713,2.58362,2.60868,2.6318,2.5474,2.5530,2.5411,2.5761];
Np = 12; % numero picchi
N = 11, % numero cicli
Tacogramma = zeros(N,1);
bpm = zeros(size(Tacogramma));
for i=1:11
    Tacogramma(i) = tempi_picchi(i+1)-tempi_picchi(i)
    bpm(i) = (1/Tacogramma(i))*60
end
Battito_medio = mean(bpm);
Battito_min = min(bpm);
Battito_max = max(bpm);

%FIGURA
figure(1)
subplot(211)
plot(time,ECG,'k',time,ECG_filtrato,'r')
xlabel('time[s]')
ylabel('segnale[mV]')
legend('segnale originale','segnale filtrato')
title('confronto tra segnale originale e filtrato')
axis tight
hold on
subplot(212)
plot(time,ECG_filtrato,'r')
title('segnale filtrato e picchi RR')
hold on
stem(tempi_picchi,picchi,'b')
xlabel('time[s]')
ylabel('segnale[mV]')
axis tight
legend('segnale filtrato','picchi R')

save Risultati.mat Durata ECG_filtrato Tacogramma N Battito_medio Battito_min Battito_max
saveas(gcf,'ECG.fig')


```

ESAME ELABORAZIONE SEGNALI BIOMEDICI - ESERCITAZIONE MATLAB

Anno Accademico 2022/2023 - PRIMO APPELLO

NOME: COGNOME:

NUMERO MATRICOLA: POSTAZIONE PC #:

Si consideri il segnale TEST01.mat, contenente il tracciato di un **ECG** (unità di misura mV) acquisito alla frequenza di Fc=500Hz e definito su una griglia temporale in cui il primo campione viene associato al tempo t=0 secondi.

PARTE 1 – COSTRUZIONE DEL SEGNALE DI INGRESSO (2pt)

Dalla registrazione originale si estraggano gli N=1000 elementi, a partire dal tempo di acquisizione t=0.5 secondi. Associare tale vettore di elementi al vettore *segnale_originale*, e definire un nuovo vettore *time*, rappresentante la griglia temporale di tale segnale. Dato il *segnale_originale*, calcolare i valori associati alla media del segnale e al suo range max e minimo e completare la seguente tabella

VARIABILE	RISPOSTA	UNITÀ DI MISURA
Numero di campioni *segnale_originale*	1000	n/a
Durata temporale del *segnale_originale*	1.998	s
Intervallo temporale [tstart,tend] del *segnale_originale*	[0.5, 2.4998]	s
Media del *segnale_originale*	0.0612	mV
Minimo del *segnale_originale*	-0.1550	mV
Massimo del *segnale_originale*	0.89	mV

PARTE 2 – FILTRAGGIO (2pt)

Progettare un filtro passa basso con due poli di modulo 0.95 alla frequenza di taglio di 5 Hz, uno zero nell'origine e uno zero sul cerchio di raggio unitario alla frequenza di 0 Hz. Imporre al filtro un guadagno G=0.1. Calcolare la variabile *segnale_filtrato* rappresentante l'uscita del filtro applicato alla variabile *segnale_originale*.

VARIABILE	RISPOSTA	UNITÀ DI MISURA
Numero di campioni *segnale_filtrato*	1000	n/a
Durata temporale del *segnale_filtrato*	1.998	s
Intervallo temporale [tstart,tend] del *segnale_filtrato*	[0.5, 2.4998]	s

PARTE 3 – ANALISI SPETTRALE (2pt)
Dati il *segnale_originale* e il *segnale_filtrato* utilizzare il metodo del Peridiogramma per calcolare la densità spettrale di potenza dei due segnali. Calcolare la densità spettrale di potenza media dei due segnali nell'intervallo di frequenza 0-10Hz e completare la seguente tabella

	RISPOSTA	UNITÀ DI MISURA
Densità spettrale di potenza media *segnale originale*	0.419	mV^2/Hz

FILE DA CONSEGNARE

- FILE SCRIPT Cognome_Nome_Matricola_ESAME.m contenente lo script utilizzato per risolvere esame
- FILE FIGURA (1pt) Cognome_Nome_Matricola_SEGNALE.fig che confronta in un unico plot il segnale_orginale (plot in colore blu) con il segnale_filtrato (plot in colore rosso)
- FILE FIGURA (1pt) Cognome_Nome_Matricola_FILTRO.fig che riporta il digramma poli/zeri del filtro
- FILE FIGURA (1pt) Cognome_Nome_Matricola_SPETTRO.fig che confronta su un unico plot il peridiogramma del segnale originale (plot in colore blu) e del segnale_filtrato (plot in colore rosso)
- FILE MATLAB (3pt) Cognome_Nome_Matricola_RISULTATI.mat che riporta le seguenti variabili:

    - segnale_originale: vettore 1-D che rappresenta il segnale originale
    - segnale_filtrato: vettore 1-D che rappresenta il segnale filtrato
    - time: vettore 1-D che rappresenta i tempi del segnale originale e filtrato
    - z: vettore 1-D che rappresenta gli zeri del filtro
    - p: vettore 1-D che rappresenta i poli del filtro
    - modulo: vettore 1-D che rappresenta il modulo del filtro nell'intervallo di frequenze [0 250 Hz] definito su N=2048 punti
    - fase: vettore 1-D che rappresenta la fase del filtro nell'intervallo di frequenze [0 250 Hz] definito su N=2048 punti ed espressa in gradi

```
clc
clear all
close all

%% OPENING
load TEST01.mat

%setting
N = length(ECG);
Fc = 500;
T = 1/Fc;
t = 0:T:T*(N-1);

%% DOMANDA 1 - CROPPING
tstart = 0.5;
tend = (N-1)*T+0.5; %tcropping 1000 samples
ind = intersect(find(t>=tstart),find(t<=tend));
time = t(ind);
segnale_originale = ECG(ind);

media_segnale_originale = mean(segnale_originale);
min_segnale_originale = min(segnale_originale);
max_segnale_originale = max(segnale_originale);

%% DOMANDA 2 - FILTRAGGIO
% Definisco poli
F = 5;
theta=(2*pi/Fc)*F;
poli = [0.95*exp(1i*theta), 0.95*exp(-1i*theta)];
%Definisco zeri
zeri = [0 1];

b = poly(zeri);
a = poly(poli);
%Imposto guadagno unitario a 250 Hz
G=1/10;
b=b*G;

%filtro
[H,F]=freqz(b,a,2048,Fc);

modulo = abs(H);
fase = angle(H)*360/(2*pi); %Definita in gradi

%segnale filtrato
segnale_filtrato = filter(b,a,segnale_originale);
media_segnale_filtrato = mean(segnale_filtrato);
min_segnale_filtrato = min(segnale_filtrato);
max_segnale_filtrato = max(segnale_filtrato);

%% DOMANDA 3 - SPETTRO
Ns = 1000; %NO ZERO PADDING
FTx=fft(segnale_originale,Ns);
S=(abs(FTx).^2)/Ns;
FTx_filtrato =fft(segnale_filtrato,Ns);
S_filtrato=(abs(FTx_filtrato).^2)/Ns;
f_FT=(0:Fc/Ns:Fc-Fc/Ns);

% POWER 0-10Hz
fstart = 0;
fend = 10;
indF = intersect(find(f_FT>=fstart),find(f_FT<=fend));
potenza_originale=mean(S(indF));
potenza_filtrato=mean(S_filtrato(indF));

```

tend = (1000-1)*T+0,5

```
67 %% FIGURE
68 % SEGNALI
69 plot(time,segnale_originale);
70 hold on
71 plot(time,segnale_filtrato,'r')
72 title('SEGNALI')
73 xlabel('Tempo (secs)')
74 ylabel('ECG (mV)')
75 legend('Segn. Originale','Segn Filtrato')
76
77 % ZP figure
78 figure
79 zplane(b,a)
80 title('Diagramma ZERI/POLI');
81
82 % Diagramma di Bode del filtro
83 figure
84 subplot(2,1,1)
85 plot(F,modulo)
86 title('modulo')
87 xlabel('Hz')
88 ylabel('mV/Hz')
89 subplot(2,1,2)
90 plot(F,fase)
91 title('fase')
92 xlabel('Hz')
93 ylabel('Degree')
94
95 %SPETTRO
96 figure
97 plot(f_FT(1:Ns/2),S(1:Ns/2))
98 hold on
99 plot(f_FT(1:Ns/2),S_filtrato(1:Ns/2),'r')
100 title('Peridiogramma')
101 ylabel('PSD [mV^2/Hz]')
102 xlabel('Hz')
103 legend('Segn. Originale','Segn Filtrato')
```

ESAME ELABORAZIONE SEGNALI BIOMEDICI - ESERCITAZIONE MATLAB

Anno Accademico 2022/2023 - SECONDO APPELLO

NOME: COGNOME:

NUMERO MATRICOLA: POSTAZIONE #:

Si consideri il segnale TEST02.mat, contenente il tracciato di un **EMG** (unità di misura mV) acquisito alla frequenza di Fc=500Hz e definito su una griglia temporale in cui il primo campione viene associato al tempo t=0 secondi.

PARTE 1 – COSTRUZIONE DEL SEGNALE DI INGRESSO (4pt)

Dalla registrazione EMG (che rappresenta il *segnale_originale*) si estraggano i cinque eventi di contrazioni della durata di 2 secondi ciascuno a partire dagli istanti tempoTaglio= [2, 6, 10, 14, 18] secondi (Nota: il segnale nei tempi di taglio, va considerato segnale utile nell'analisi della contrazione). Calcolare i valori associati alla media del segnale EMG nelle cinque contrazioni e al suo range max e minimo e completare la seguente tabella

VARIABILE	RISPOSTA	UNITÀ DI MISURA
Numero di campioni di ogni epoca	1001	
Media del segnale EMG durante le cinque contrazioni	-0.0092	mV
Minimo del segnale EMG durante le cinque contrazioni	-0.8618	mV
Massimo del segnale EMG durante le cinque contrazioni	0.7538	mV

PARTE 2 – FILTRAGGIO

Progettare un filtro passa basso con due poli di modulo 0.95 alla frequenza di taglio di +/-50 Hz, un polo di modulo 0.95 alla frequenza di 0 Hz, e tre zeri sul cerchio di raggio unitario alla frequenza 125, 250 e 375 Hz. Imporre al filtro un guadagno unitario alla frequenza nulla. Calcolare la variabile *segnale_filtrato* rappresentante l'uscita del filtro applicato al *segnale_originale*.

PARTE 3 – ANALISI SPETTRALE (2pt)

Dati il *segnale_originale* e il *segnale_filtrato* utilizzare il metodo del Peridiogramma per calcolare la densità spettrale di potenza del segnale (Nota implementativa: no zero padding, utilizzare il numero di campioni dei due segnali). Calcolare la densità spettrale di potenza media dei due segnali nell'intervallo di frequenza 0-50Hz e completare la seguente tabella

	RISPOSTA	UNITÀ DI MISURA
Densità spettrale di potenza media segnale originale	0.0266	mV^2/Hz

FILE DA CONSEGNARE

- FILE SCRIPT Cognome_Nome_Matricola_ESAME.m contenente lo script utilizzato per risolvere esame
- FILE FIGURA (1pt) Cognome_Nome_Matricola_SEGNALE.fig che confronta in un unico plot il segnale_orginale (plot in colore blue) con il segnale_filtrato (plot in colore rosso)
- FILE FIGURA (1pt) Cognome_Nome_Matricola_FILTRO.fig che riporta il digramma poli/zeri del filtro
- FILE FIGURA (1pt) Cognome_Nome_Matricola_BODE.fig che riporta il modulo semplice (NO CONVERSIONE IN DB) e la fase semplice del filtro (NO CONVERSIONE IN GRADI)
- FILE FIGURA (1pt) Cognome_Nome_Matricola_SPETTRO.fig che confronta su due panelli affiancati il peridiogramma del segnale originale e quello del sanle filtrato
- FILE MATLAB (4pt) Cognome_Nome_Matricola_RISULTATI.mat che riporta le seguenti variabili:

    - segnale_originale: vettore 1-D che rappresenta il segnale originale
    - segnale_filtrato: vettore 1-D che rappresenta il segnale filtrato
    - time: vettore 1-D che rappresenta i tempi del segnale originale e filtrato
    - z: vettore 1-D che rappresenta gli zeri del filtro
    - p: vettore 1-D che rappresenta i poli del filtro
    - modulo: vettore 1-D che rappresenta il modulo semplice del filtro nell'intervallo di frequenze [0 250 Hz] definito su N=2048 punti (NO CONVERSIONE IN DB)
    - fase: vettore 1-D che rappresenta la fase del filtro nell'intervallo di frequenze [0 250 Hz] definito su N=2048 punti (NO CONVERSIONE IN GRADI)
    - P_segnale_originale: vettore 1-D che rappresenta il peridiogramma del segnale originale nell'intervallo di frequenze [0 250 Hz]
    - P_segnale_filtrato: vettore 1-D che rappresenta il peridiogramma del segnale filtrato nell'intervallo di frequenze [0 250 Hz]

```m
clc
clear all
close all

%% OPENING
%load data
load TEST02.mat

%setting
N = length(EMG);
Fc = 500;
T = 1/Fc;
time = 0:T:T*(N-1);
segnale_originale=EMG;

%% DOMANDA 1 - CROPPING SEGNALE ORIGINALE
tTaglio= [2,6, 10, 14, 18]; %secondi
durata = 2; %secondi
Nepoche = length(tTaglio);

%analisi epoche originali
for i=1:Nepoche
    tStart = tTaglio(i);
    tEnd = tStart + durata;
    ind = intersect(find(time>=tStart),find(time<=tEnd));
    epoche(i).ind = ind;
    epoche(i).segnale = segnale_originale(ind)';
end
media_segnale_originale = mean([epoche.segnale]);
min_segnale_originale = min([epoche.segnale]);
max_segnale_originale = max([epoche.segnale]);

%% DOMANDA 2 - FILTRAGGIO
% Definisco poli
F = 50;
theta=(2*pi/Fc)*F;
poli = [0.95*exp(1i*theta), 0.95*exp(-1i*theta) 0.95];
%Definisco zeri
zeri = [+1i -1i -1];

%Progetto il filtro e definisco il guadagno
b = poly(zeri);
a = poly(poli);
G=polyval(b,1)/polyval(a,1);
b=b/G;
segnale_filtrato = filter(b,a,segnale_originale);


%% DOMANDA 3 - SPETTRO
FTx=fft(segnale_originale,N);
S=(abs(FTx).^2)/N;
P_segnale_originale = S(1:N/2);
FTx_filtrato =fft(segnale_filtrato,N);
S_filtrato=(abs(FTx_filtrato).^2)/N;
P_segnale_filtrato = S_filtrato(1:N/2);
f_FT=(0:Fc/N:Fc-Fc/N);

% POWER 0-50Hz
fstart = 0;
fend = 50;
indF = intersect(find(f_FT>=fstart),find(f_FT<=fend));
potenza_originale=mean(S(indF));
potenza_filtrato=mean(S_filtrato(indF));

```

```
66
67 %% FIGURE
68
69 % SEGNALI
70 figure
71 plot(time,EMG);
72 hold on
73 plot(time,segnale_filtrato,'r')
74 xlabel('Time (secs)')
75 ylabel('EMG (mV)')
76 title('CONFRONTO SEGNALI')
77 legend('Segnale Originale','Segnale Filtrato')
78
79 % ZP figure
80 figure
81 zplane(b,a)
82
83 % Diagramma di Bode del filtro
84 figure
85 [H,F]=freqz(b,a,2048,Fc);
86 subplot(2,1,1)
87 plot(F,abs(H))
88 title('modulo')
89 xlabel('Hz')
90 subplot(2,1,2)
91 plot(F,angle(H))
92 title('fase')
93 xlabel('Hz')
94
95 %PSD
96 figure
97 plot(f_FT(1:N/2),S(1:N/2))
98 hold on
99 plot(f_FT(1:N/2),S_filtrato(1:N/2),'r')
100 xlabel('Hz')
101 ylabel('PSD')
102 title('Power Spectral Density [0-250Hz]')
103 legend('Segnale Originale','Segnale Filtrato')
104
```