

Introduzione all' uso di Matlab

Elaborazione di Segnali e Dati Biomedici

Anno Accademico 2017-2018

Prof. Mario Cesarelli



Istruzione LOAD

È possibile salvare su disco o caricare da disco variabili (scalari, vettori, matrici)

LOAD FILENAME X permette di caricare da un file la sola variabile X

LOAD FILENAME X Y Z permette di caricare da un file le variabili X Y Z

L'istruzione LOAD usa un file binario tipo Mat-file

Per caricare dati da file ascii (file testo editabili) usare

LOAD -ASCII FILENAME X Y Z



Istruzione UIGETFILE

In alternativa è possibile caricare da qualsiasi directory il file di interesse con i seguenti comandi:

```
>> [nomefile, pathfile] = uigetfile('*.*asc', 'carica il segnale ecg'); % (vedere help uigetfile)
ecg = load ([pathfile nomefile]); % carica il file
```

Name	Value
ecg	6143x3 double
nomefile	'ecg.asc'
pathfile	'C:\Users\ ...'



Istruzione ZEROS

$X = \text{zeros}(n, m);$

L'istruzione *zeros* del Matlab permette di creare una matrice “n x m” di zeri.

Per esempio:

```
>> x = zeros(2,3)
```

```
x =
```

```
0     0     0
0     0     0
```



Istruzione ONES

$X = \text{ones}(n, m)$

L'istruzione *ones* del Matlab permette di creare una matrice “n x m” di uno.

Per esempio:

```
>> x = ones(2, 3)
```

```
x =
```

```
    1    1    1
    1    1    1
```



Istruzione if-else

```
Command Window
>> limit = 0.75;
A = rand(10,1)

A =

    0.7513
    0.2551
    0.5060
    0.6991
    0.8909
    0.9593
    0.5472
    0.1386
    0.1493
    0.2575

>> if any(A > limit)
    disp('There is at least one value above the limit.')
else
    disp('All values are below the limit.')
end
There is at least one value above the limit.
fx >>
```



Istruzione GINPUT

[x,y] = ginput (n)

L'istruzione *ginput* del Matlab consente di determinare, tramite mouse, le coordinate di n punti del grafico.

Il numero tra parentesi (n) rappresenta quanti "clic" Matlab si aspetta prima di tornare al workspace.

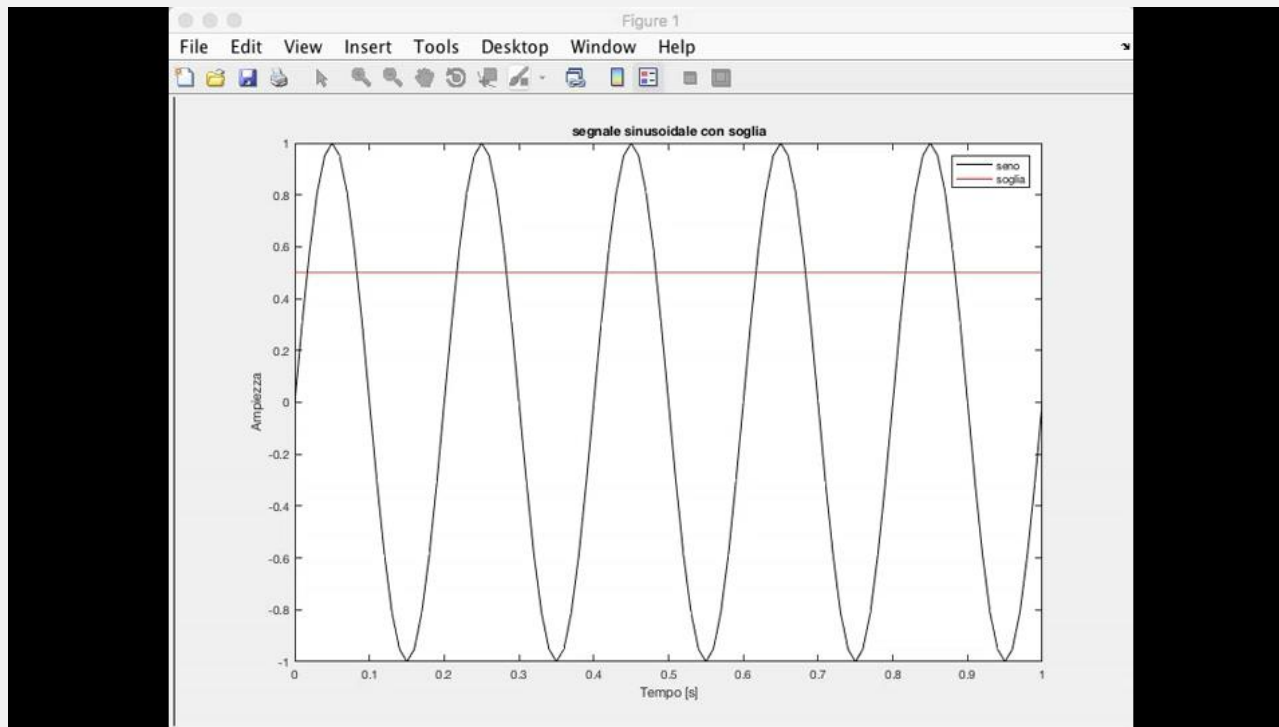
Qualora (n) non sia specificato, Matlab acquisisce i campioni in maniera continuativa, finché l'utente non preme "invio".



Istruzione GINPUT

[tempi, ampiezze] = ginput (2)

% acquisisco 2 campioni



Istruzione GINPUT

[tempi,ampiezze] = ginput (2)

% acquisisco 2 campioni

tempi =

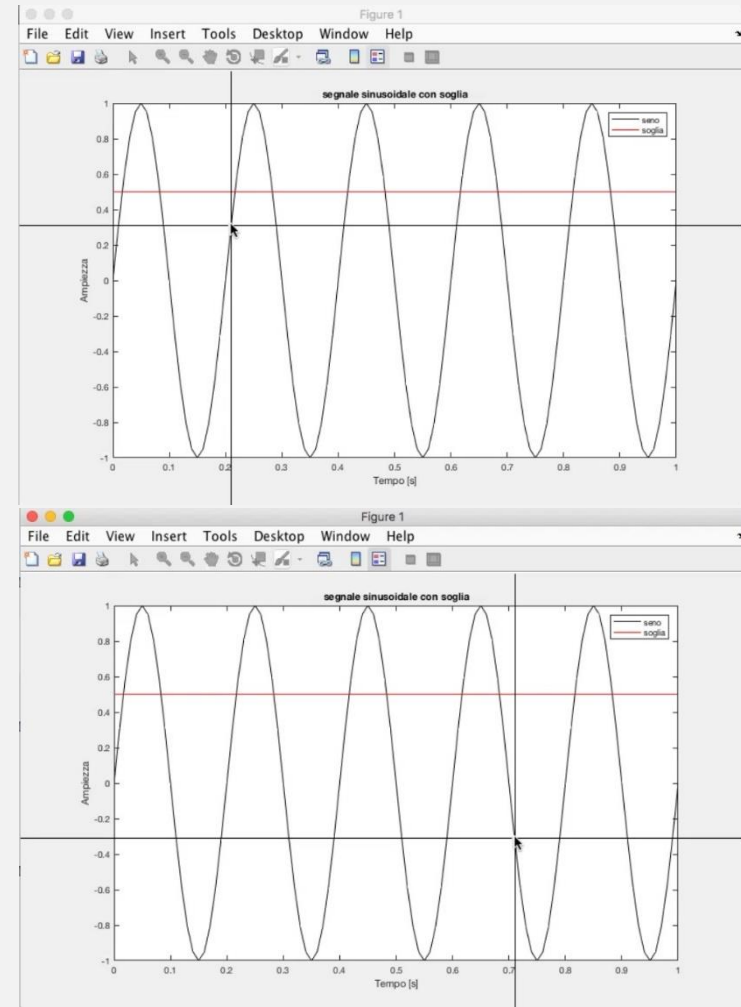
0.2099

0.7010

ampiezze =

0.2998

-0.0234



Istruzione SAVE

save **Esercitazione_3** tempi ampiezze

Creo una variabile Esercitazione_3.mat nel current folder, in cui vengono registrate le variabili specificate.



Creazione soglia di riferimento

```

Fs=100; %freq camp
Ts=1/Fs; % periodo camp
t = [0:Ts:1]; %asse dei tempi in s

```

```
% generazione sinusoide
```

```

f = 5;
x = sin(2*pi*t*f);

```

```
%creazione soglia
```

```
soglia = 0.5 * ones (length(t),1);
```

```
figure %creo una nuova figura
```

```
plot(t,x,'k')
```

```
hold % mantengo la figura corrente e sovrappongo il plot successivo
```

```
plot(t,soglia,'r')
```

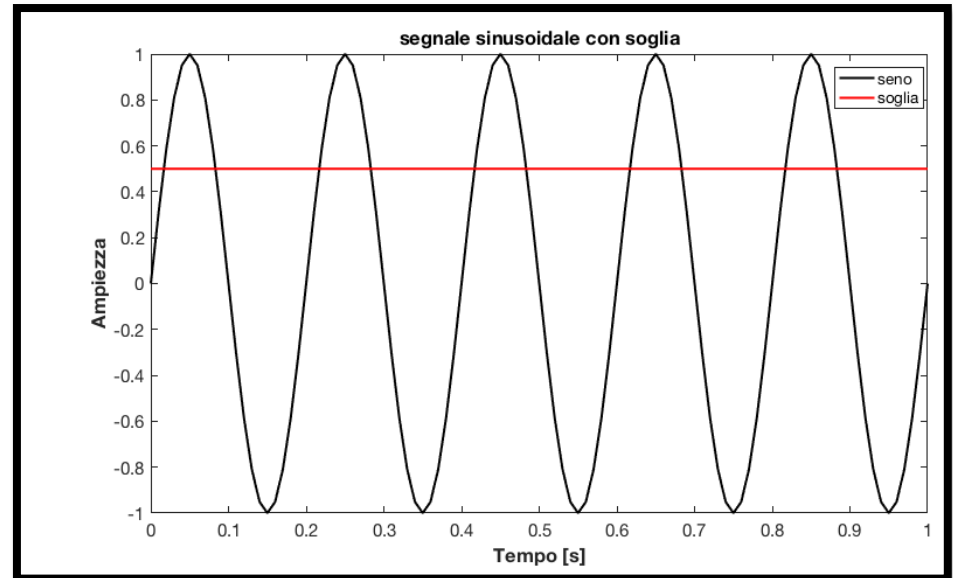
```
title('segnale sinusoidale con soglia')
```

```
xlabel('Tempo [s]')
```

```
ylabel('Ampiezza')
```

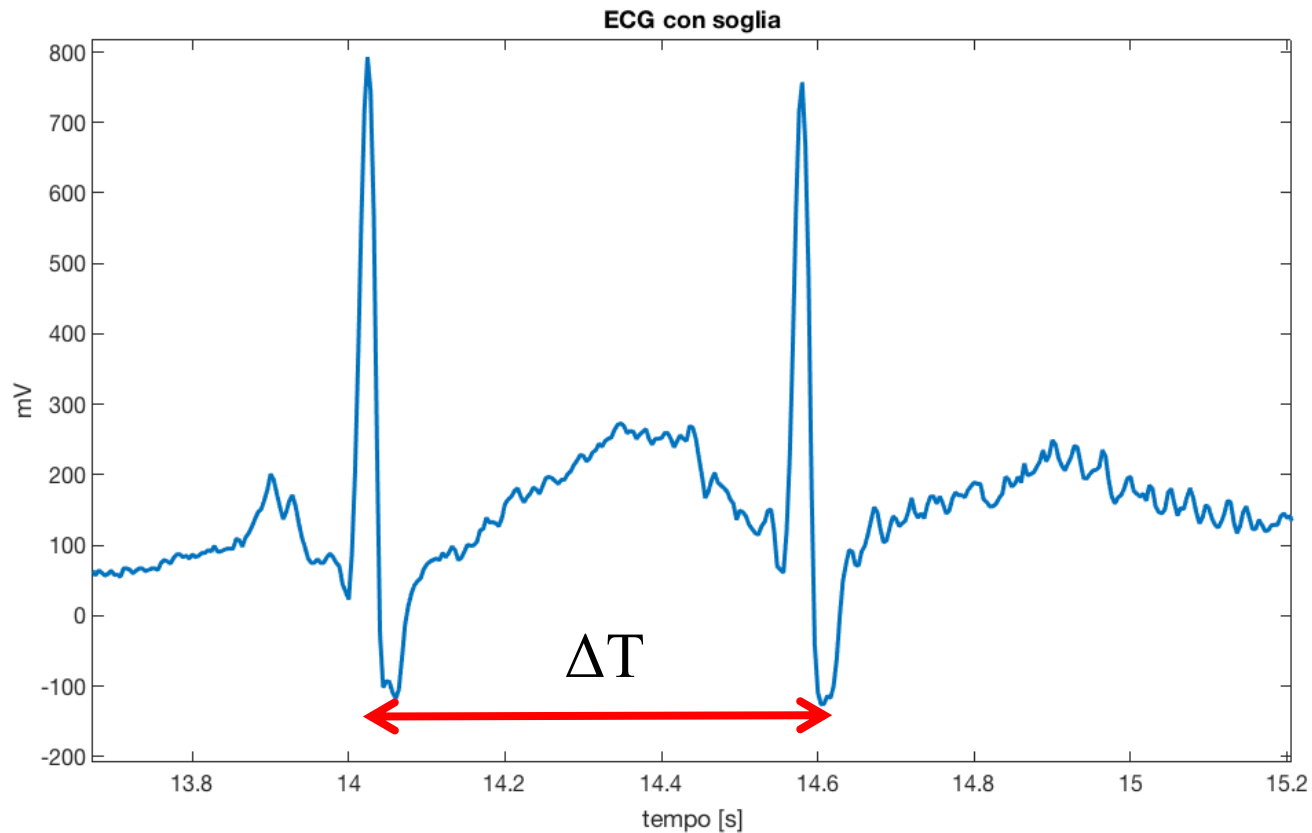
```
legend('seno','soglia')
```

```
axis([t(1) t(end) min(x) max(x)])
```



Frequenza istantanea ECG

$$F_{\text{ist}} = 60/\Delta T \text{ [bpm]}$$



Finestre di scorrimento sul segnale

passo=10; % lunghezza della finestra in campioni

fine=5*passo;% termine scorrimento

figure

for i = 1:passo:fine

plot(t(i:i+passo-1),x(i:i+passo-1))

axis([t(i) t(i+passo-1) min(x) max(x)])

title('finestra temporale')

xlabel('tempo [s]')

ylabel('Ampiezza')

pause

end

