

◆ *MATLAB: linguaggio di programmazione*

L'ambiente MATLAB possiede un completo linguaggio di programmazione. Vediamo la sintassi delle espressioni che permettono di controllare il flusso dell'esecuzione (cicli, salti condizionali):

```
for var=[insieme di valori],  
    azioni...  
end,
```

Ad ogni ripetizione del ciclo la variabile assume un valore estratto dal vettore [insieme di valori]: non è necessario che questo vettore sia equispaziato o costruito da soli valori interi!!

```
while (espressione logica)  
    azioni...  
end
```

```
if (espressione logica)  
    azioni...  
else  
    azioni...  
end,
```

Gli operatori logici in MATLAB sono:

==	uguale
~=	diverso
<, >, <=, >=	disuguaglianze
&	AND logico
	OR logico

☒ *Esercizio*

Si vuole scrivere una funzione matlab (**even.m**), che, dato in input un vettore (matrice) costituito da numeri interi, restituisca 1 in corrispondenza di numeri pari, 0 altrimenti.

```
function y=even(x),  
  
l=length(x(:));          % lunghezza del vettore  
y=zeros(size(x));  
  
for k=1:l,  
    if (floor(x(k)/2)==x(k)/2),  
        y(k)=1;  
    end,  
end,
```

ATTENZIONE: la funzione precedente è corretta anche se l'input è una matrice! Infatti esiste un modo alternativo per riferirsi ad un elemento in posizione (riga,colonna): utilizzando un solo indice, si contano gli elementi progressivamente lungo le colonne a partire dalla prima.

```
A = [11 3 2; 5 1 11; 9 6 9; 2 15 14]
```

```
A(3,2)=6
```

```
A(7)=6
```

La funzione **find()** restituisce questo tipo di indice.

Si riscriva in modo compatto la funzione **even()**, utilizzando la funzione **find**.

☒ *Esercizio*

Si vuole scrivere una funzione matlab (**ondaquadra.m**), che, restituisca in uscita l'ordinata dell'onda quadra di semiperiodo T, rispetto al vettore delle ascisse x (si utilizzi la funzione **even()** appena sviluppata).

```
function y=ondaquadra(x,T),
```

```
y=-ones(size(x)); % ones(r,c) crea una matrice inizializzata a 1  
set=find(even(floor(x/T))==1);  
y(set)=1;
```

◆ *MATLAB: visualizzazione*

Ogni istruzione grafica viene eseguita nella finestra attiva. Per creare più finestre, si utilizza il comando **figure**, che crea una nuova finestra (e la rende attiva).

Per fissare gli estremi di ascissa e ordinata mostrati nella finestra grafica, si utilizza il comando:

```
axis([xmin xmax ymin ymax])
```

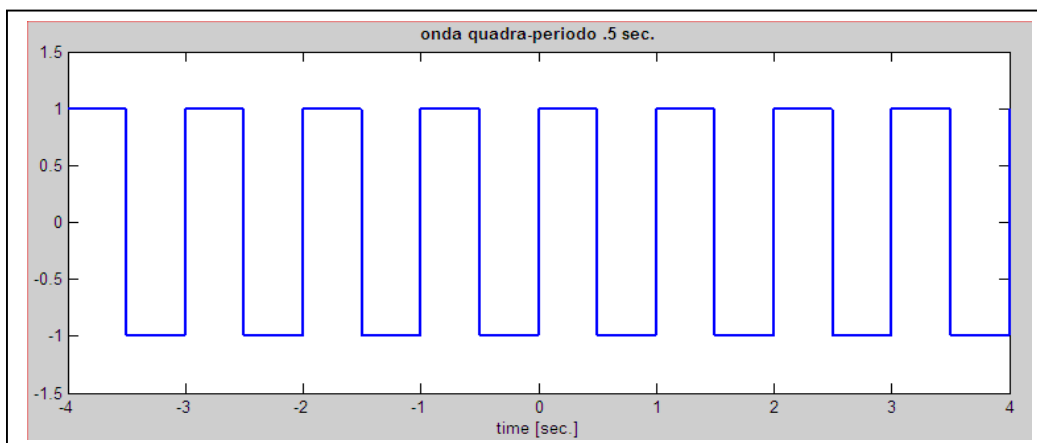
Infine per aggiungere un testo agli assi e alla figura stessa, esistono i comandi:

```
xlabel('testo'), ylabel('testo'), title('testo')
```

ATTENZIONE: le stringhe in MATLAB sono delimitate da apici.

Si generi e si visualizzi, ad esempio, un'onda quadra di periodo 0.5 secondi.

```
» t=[-4:.004:4];  
» s=ondaquadra(t,.5);  
» plot(t,s);  
» axis([-4 4 -1.5 1.5]);  
» xlabel('time [sec]');  
» title('onda quadra-semiperiodo .5 sec');
```



◆ *Convoluzione di segnali in MATLAB*

Si scriva la funzione `convolve.m` che permette di convolvere due segnali (memorizzati in vettori riga) tra loro. In realtà MATLAB possiede già un comando per effettuare la convoluzione di sequenze `conv()`.

```
function X=convolve(A,B);

la=length(A);
lb=length(B);

X=zeros(1,la+lb-1);    % lunghezza del segnale risultante

B=fliplr(B);          % ribalta il segnale B

for k=1:(la+lb-1),
    tmp2=[zeros(1,k-1),B,zeros(1,la+lb-k-1)];
    X(k)=sum(A.*tmp2(lb:la+lb-1));
end
```

ATTENZIONE: la durata del segnale risultante è data dalla somma delle durate dei segnali in ingresso! Nel caso di segnali approssimati da sequenze l'effettiva durata è $(la+lb-1)$ con la , lb , durata dei segnali in ingresso espressa in campioni. Nella realizzazione numerica dell'integrale (somma) **manca** anche il fattore di scala dt , che si può inserire moltiplicando il risultato della convoluzione per il passo di campionamento, come mostrato nei prossimi esempi.

La convoluzione di segnali continui verrà quindi simulata in MATLAB come:

```
C=conv(A,B)*dt;
```

Ora si deve associare al segnale C l'asse dei tempi tc corretto. Il passo di discretizzazione è sempre dt (comune ai segnali A, B, C). L'ordinata memorizzata nel primo campione di C, $C(1)$ corrisponde all'istante $ta(1)+tb(1)$ (con ta e tb assi dei tempi su cui sono definiti i segnali A e B). L'asse dei tempi tc associato al risultato della convoluzione C si può scrivere come

```
» tc=(ta(1)+tb(1))+[0:length(C)-1]*dt;
```

Negli esempi seguenti, poiché entrambi i segnali A e B sono definiti a partire dal tempo $t=0$ sec, la definizione di tc si può semplificare:

```
» tc=[0:length(C)-1]*dt;
```

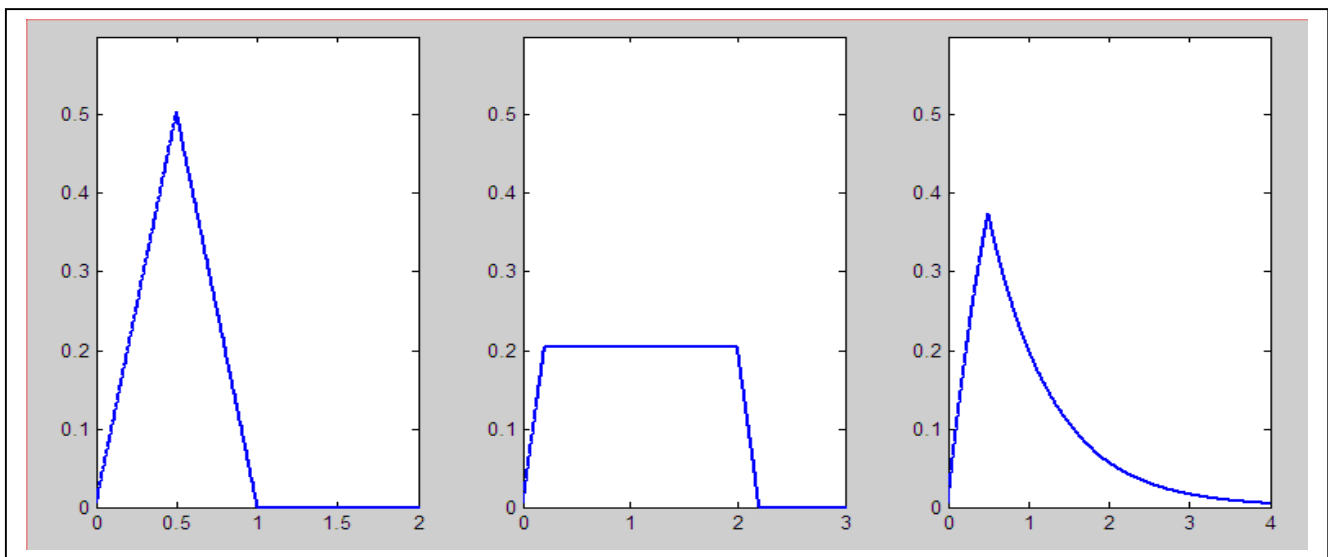
☒ *Esercizio 1*

Si calcoli e si visualizzi la convoluzione tra:

- Due rettangoli (resi causali) di durata 0.5 secondi
- Un rettangolo di durata 0.2 sec. e un rettangolo di durata 2 sec. (entrambi resi causali)
- Un rettangolo (causale) di durata 0.5 sec. e un segnale esponenziale $s(t)=\exp(-t/T_0)u(t)$, $T_0=0.8$ sec

Si utilizzi come asse temporale il vettore:

t=[0:0.004:4];

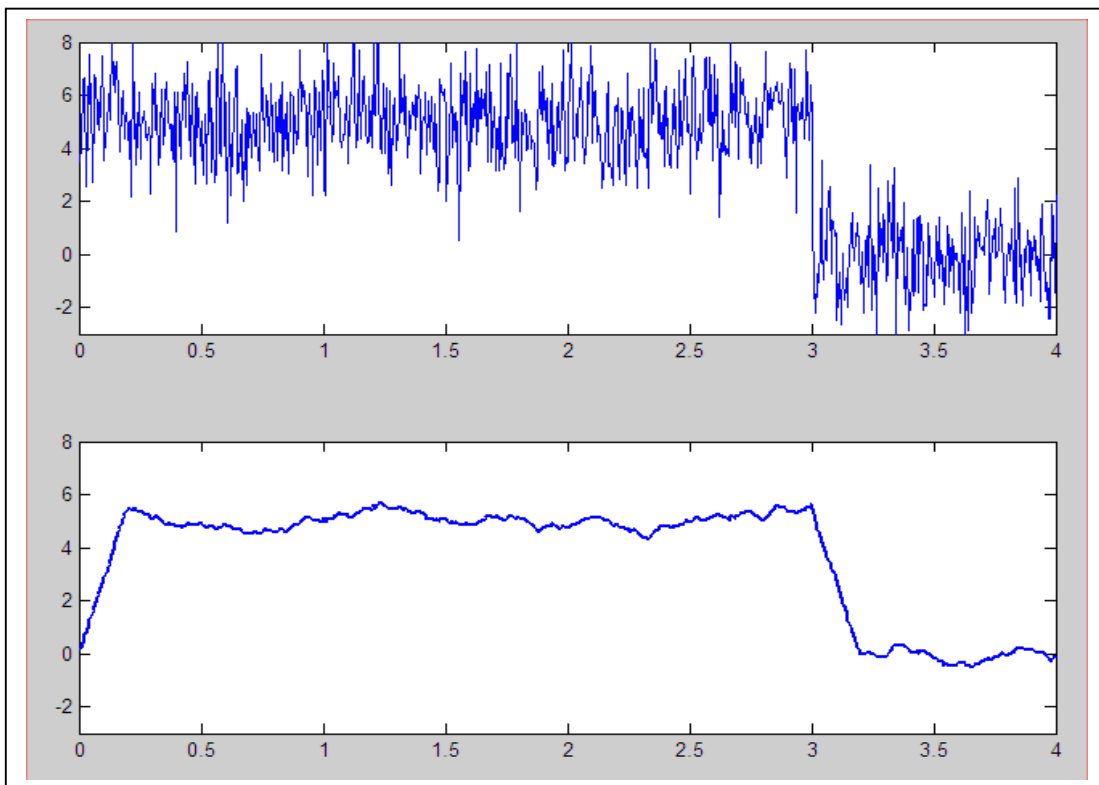


◆ *Convoluzione: filtraggio di segnali*

L'operazione di filtraggio di segnale corrisponde alla convoluzione tra segnale e la risposta all'impulso del filtro scelto. Ad esempio, una risposta all'impulso rettangolare può essere pensata come un filtro a media mobile. Vediamo l'effetto dell'applicazione del filtraggio a media mobile su un segnale costante corrotto da un disturbo di natura aleatoria.

☒ *Esercizio 2*

Si generi un segnale costante $s(t)$, di durata 3 sec. e ampiezza 5, a cui viene sovrapposto un disturbo con distribuzione dei valori normale, a media $m=0$ e varianza $\sigma^2=2$. Il segnale è osservato per 4 sec., il passo di rappresentazione è $\Delta t=5$ ms. Si filtri il segnale con un filtro a media mobile (risposta impulsiva rettangolare) di durata 0.2 sec. Si visualizzino il segnale $s(t)$ ed il risultato del filtraggio. Si noti come la risposta impulsiva del filtro a media mobile sia simulata costruendo direttamente un vettore (funzione `ones()`) di lunghezza opportuna: non è necessario che i due segnali da convolvere siano definiti a partire dallo stesso asse temporale. In generale i due assi temporali (che devono però avere in comune il passo di rappresentazione!) avranno lunghezze diverse.



☒ Esercizio 3

Si generi un segnale (tempo di osservazione 4 sec, passo di campionamento $\Delta t=5$ ms) somma di 2 sinusoidi, di frequenza 5 Hz e 20 Hz (quest'ultima con ampiezza doppia rispetto alla prima). Si vuole estrarre la prima sinusoida dal segnale complessivo (filtraggio passa-basso).

- si filtri il segnale con un filtro a media mobile (rettangolo) di durata 0.1 sec. Si visualizzino il segnale $s(t)$ ed il risultato del filtraggio.
- Qual è la risposta in frequenza del filtro utilizzato? Quali sono quindi le componenti sinusoidali del segnale che questo filtro permette di eliminare? Perché?

