

MATLAB-SIMULINK

Creazione di grafici 2D e 3D

Ing. Alessandro Pisano

apisano@unica.it

```
s='set(gcf,'position',[0,0,400,350]);';
```

Parte prima

Affrontiamo come passo preliminare il problema della **visualizzazione del grafico 2D di una funzione scalare** in un intervallo di interesse $[a, b]$.

$$y = f(x) \quad x \in [a, b]$$

Partizione del dominio $[a, b]$

$$[x_1, x_2, \dots, x_N]$$
$$x_1 = a \quad x_N = b \quad x_i < x_{i+1}$$



Devono essere creati due vettori

$$\mathbf{x} = [x_1, x_2, x_3, \dots, x_N]$$

$$\mathbf{y} = [y_1, y_2, y_3, \dots, y_N] \quad y_i = f(x_i) \quad i = 1, 2, \dots, N$$

N deve essere "sufficientemente grande" affinché il grafico della funzione risulti sufficientemente regolare.

Dopo avere creato i due vettori, è possibile passarli come argomento di ingresso alla funzione **plot** per creare una curva che interpola mediante una spezzata i punti del piano aventi coordinate (x_i, y_i) .

Sviluppiamo alcuni esempi introduttivi riferendoci alla funzione $y(x) = 5 \sin(x) \cos^2(5x)$, che desideriamo plottare nell'intervallo $I = [0, 4\pi]$

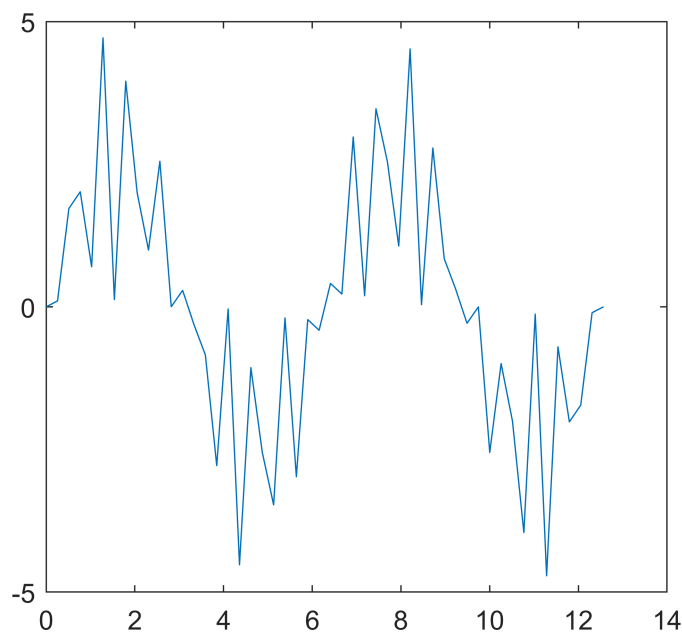
Creiamo dapprima il vettore che partiziona l'intervallo I:

Scegliamo $N = 50$.

```
N=50;  
x=linspace(0,4*pi,N);
```

Ora creiamo il vettore che contiene i valori della funzione da plottare in corrispondenza degli elementi del vettore x. Data la forma della funzione scelta è necessario utilizzare gli operatori di moltiplicazione ed elevamento a potenza **elemento per elemento**

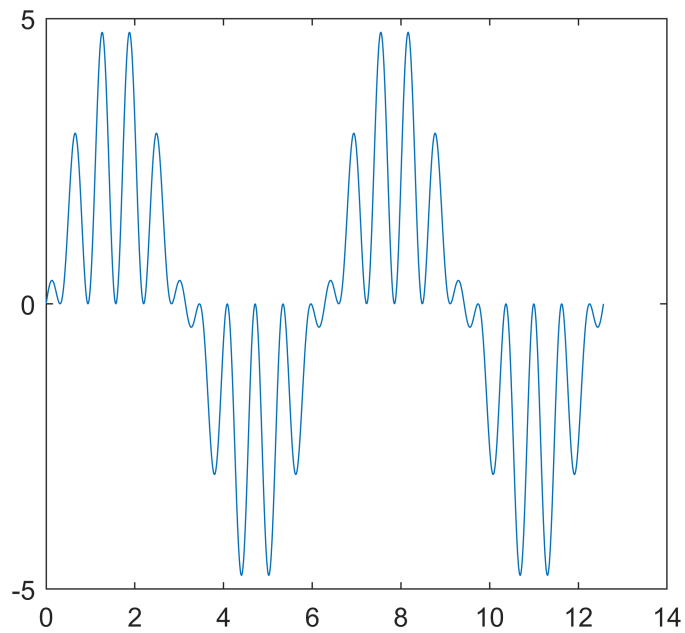
```
y=5*sin(x).*cos(5*x).^2;  
plot(x,y)  
eval(s)
```



La forma del grafico ottenuto rivela come il numero N di punti scelto (cioè la risoluzione con la quale è stato "campionato" l'intervallo $I = [0, 4\pi]$) è troppo basso. Se tale insieme di punti è troppo disperso chiaramente la curva risulta deformata.

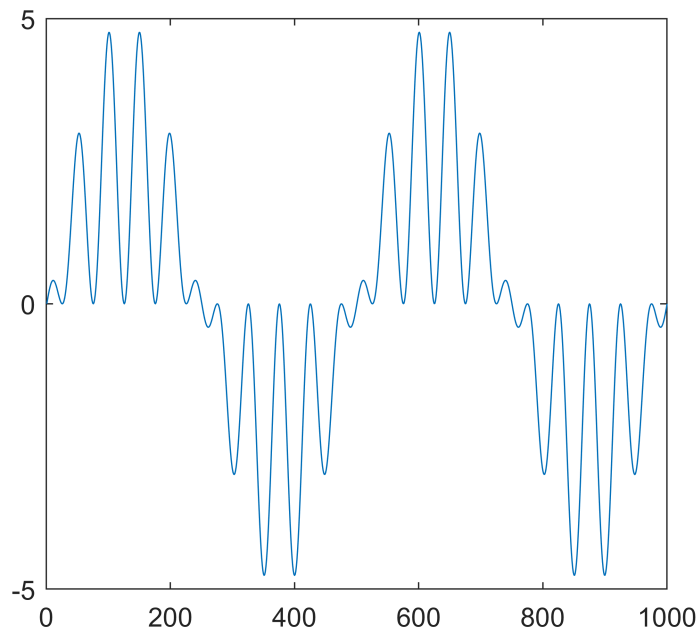
Per ottenere un grafico più regolare e fedele all'andamento reale della curva inseriamo ora $N=1000$

```
N=1000;  
x=linspace(0,4*pi,N);  
y=5*sin(x).*cos(5*x).^2;  
plot(x,y)
```



Se si passa alla funzione `plot` il solo vettore `y` dei dati, viene inserita in ascissa la posizione del generico elemento di `y`. Si veda a tal fine il seguente esempio.

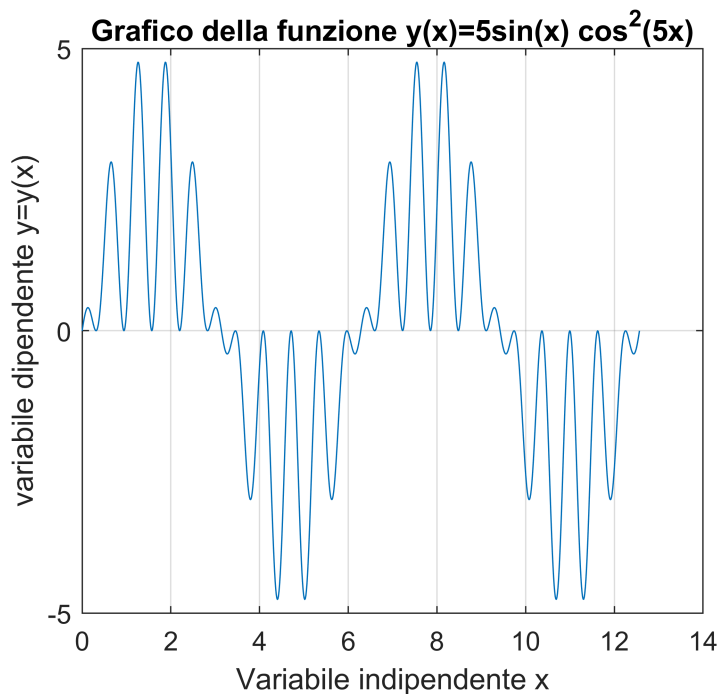
```
N=1000;
x=linspace(0,4*pi,N);
y=5*sin(x).*cos(5*x).^2;
plot(y)
```



Il grafico generato di default è spoglio e privo di etichette. Impariamo a produrre un grafico di migliore qualità ed a corredarlo di etichette descrittive appropriate.

Aggiungiamo una griglia sottostante al grafico (comando **grid**), inseriamo un titolo (comando **title**) ed inseriamo anche delle etichette di commento in prossimità degli assi delle ascisse e delle ordinate (comandi **xlabel** e **ylabel**). Le funzioni **title**, **xlabel** e **ylabel** accettano come argomento di ingresso una stringa di testo (delimitata quindi fra apici).

```
N=1000;  
x=linspace(0,4*pi,N);  
y=5*sin(x).*cos(5*x).^2;  
plot(x,y)  
grid  
title('Grafico della funzione  $y(x)=5\sin(x) \cos^2(5x)$ ')  
xlabel('Variabile indipendente x')  
ylabel('variabile dipendente  $y=y(x)$ ')
```

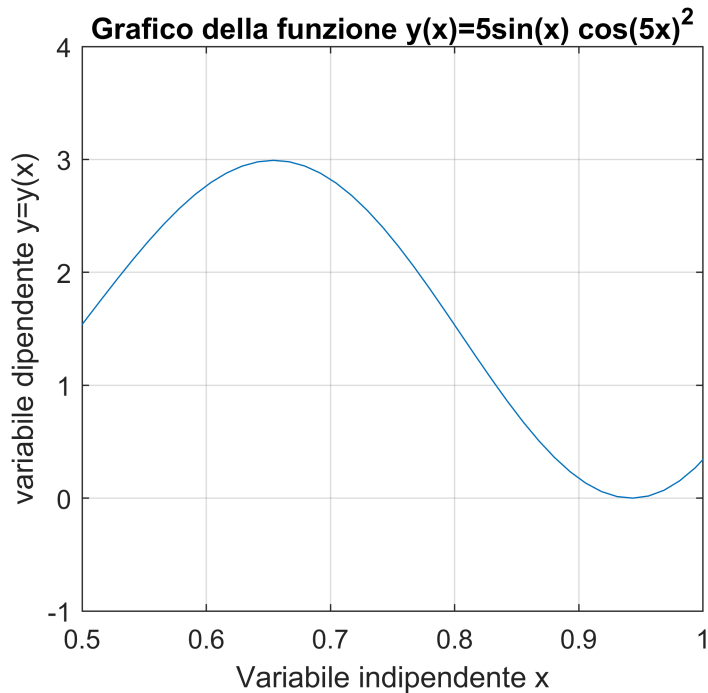


Il grafico inizia ad assumere un aspetto professionale e compatibile con l'inserimento in un documento tecnico-scientifico.

Per **ridefinire gli intervalli di visualizzazione** degli assi delle ascisse o delle ordinate si utilizzando rispettivamente i comandi **xlim** ed **ylim**

```
N=1000;  
x=linspace(0,4*pi,N);  
y=5*sin(x).*cos(5*x).^2;  
plot(x,y)  
grid  
title('Grafico della funzione  $y(x)=5\sin(x) \cos(5x)^2$ ')
```

```
xlabel('Variabile indipendente x')
ylabel('variabile dipendente y=y(x)')
xlim([0.5 1])
ylim([-1 4])
```



La sintassi

```
xlim([0.5 1])
ylim([-1 4])
```

puo essere sostituita da un unico comando che imposta simultaneamente gli intervalli di visualizzazione di entrambi gli assi coordinati.

```
axis([0.5 1 -1 4])
```

Quando, **all'interno di uno script**, la funzione `plot` viene invocata successivamente piu volte di fila essa sovrascrive più volte la medesima finestra grafica, e come risultato "sopravvive" solo il grafico generato dall'ultima istanza di chiamata alla funzione. Vediamo un esempio costruendo dei grafici a partire dai dati inseriti nella matrice `M` che contiene le letture campionate di alcuni sensori di misura. Tale matrice può essere acquisita nel workspace caricando il file `M.mat` con il comando `load`. Si esegua **in uno script esterno** (non un live script) il seguente codice di esempio

```
load M
%estraiamo dalle colonne della matrice i vettori che contengono gli istanti
%di campionamento (prima colonna) e le letture dai 5 sensori (restanti 5 colonne)

t=M(:,1);
T1=M(:,2);
P1=M(:,3);
P2=M(:,4);
T2=M(:,5);
P3=M(:,6);

%grafico sonda T1
```

```

plot(t,T1), grid
title('Lettura sonda T1')
xlabel('Tempo [s]')
%grafico sonda P1
plot(t,P1);grid
title('Lettura sonda P1')
xlabel('Tempo [s]')

```

Una possibile soluzione è anteporre a ciascuna chiamata alla funzione plot una istruzione del tipo **figure**, che pre-alloca una finestra grafica dedicata e fa sì che ciascuna chiamata alla funzione plot vada ad operare su finestre grafiche differenti. Testare il seguente codice in uno script esterno.

```

load M
%estraiamo dalle colonne della matrice i vettori che contengono gli istanti
%di campionamento (prima colonna) e le letture dai 5 sensori (restanti 5 colonne)
t=M(:,1);
T1=M(:,2);
P1=M(:,3);
P2=M(:,4);
T2=M(:,5);
P3=M(:,6);

figure
plot(t,T1), grid
title('Lettura sonda T1')
xlabel('Tempo [s]')
figure
plot(t,P1), grid
title('Lettura sonda P1')
xlabel('Tempo [s]')

```

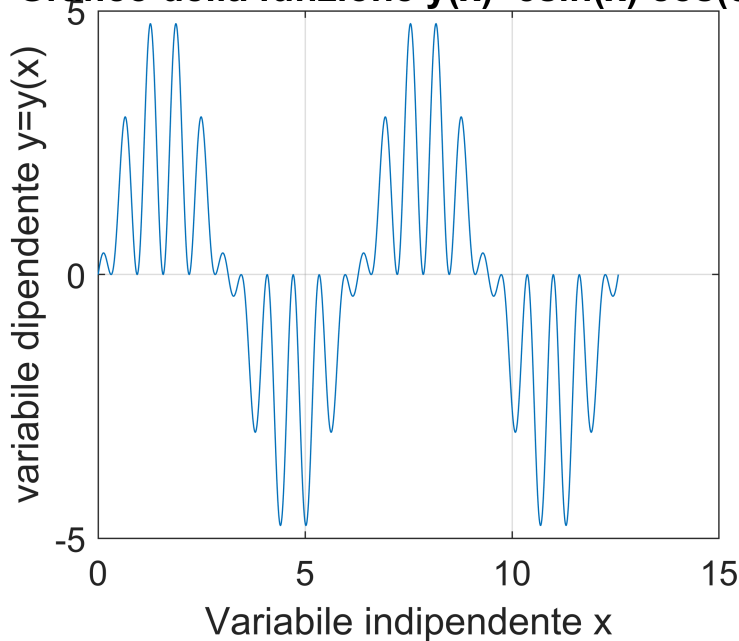
Ora completiamo ulteriormente li set di istruzioni di personalizzazione del grafico in modo da poter settare a piacimento la **dimensione dei font** che compaiono nelle label e la dimensione delle cifre riportate negli assi coordinati. Cio puo essere utile se la figura deve essere inserita in un documento tecnico con formato rimpicciolito

```

N=1000;
x=linspace(0,4*pi,N);
y=5*sin(x).*cos(5*x).^2;
dimfont=13;
plot(x,y)
grid
title('Grafico della funzione  $y(x)=5\sin(x)\cos(5x)^2$ ', 'FontSize',dimfont)
xlabel('Variabile indipendente x', 'FontSize',dimfont)
ylabel('variabile dipendente  $y=y(x)$ ', 'FontSize',dimfont)
set(gca, 'FontSize',dimfont)

```

Grafico della funzione $y(x)=5\sin(x) \cos(5x)^2$



Un grafico generato in Matlab può essere esportato in vari formati grafici con il comando "Save as" (menu "File"), oppure copiato (Edit->Copy Figure) ed incollato direttamente all'interno di un documento word.

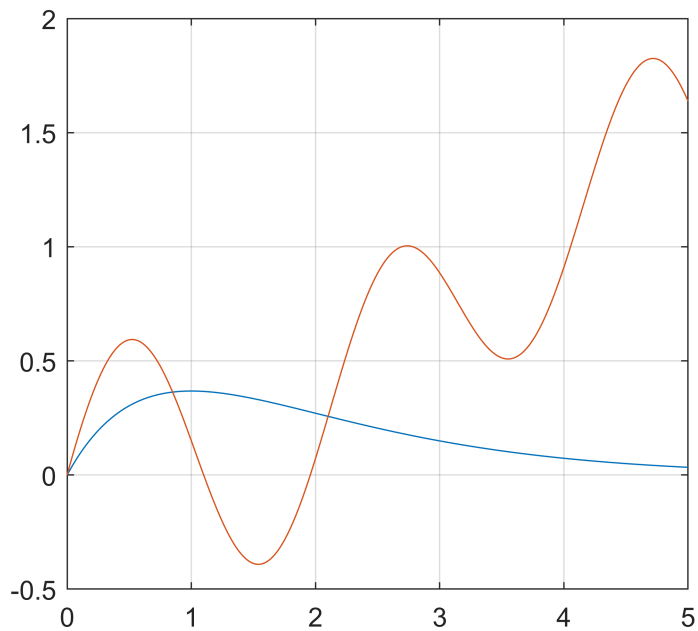
Il formato di esportazione .fig consente la riapertura, e la successiva modifica, della immagine all'interno di Matlab.

Creazione di grafici sovrapposti

La sintassi di base per la creazione di grafici con più curve sovrapposte prevede di passare come argomenti di ingresso alla funzione plot **più coppie di vettori**, una coppia per ciascuna curva da includere nel grafico.

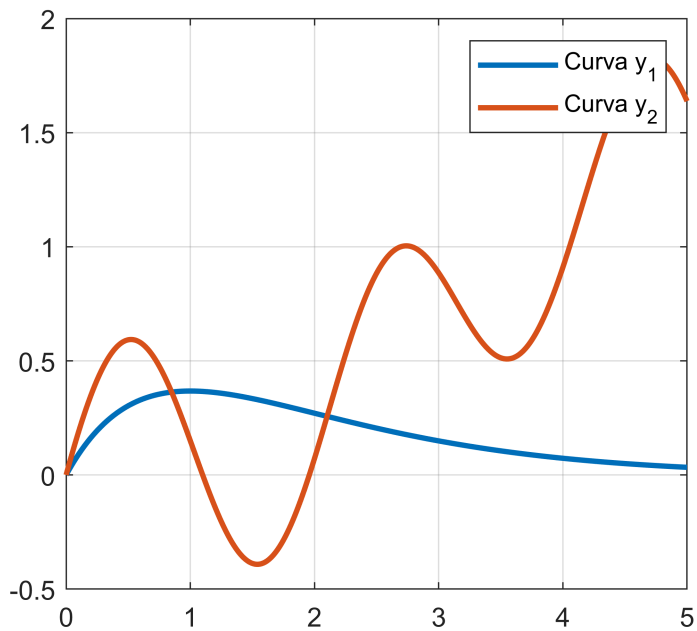
Testiamo tale funzionalità con riferimento alle due curve $y_1(x) = x e^{-x}$ ed $y_2(x) = \frac{x}{1+x} + \sin(x)\cos(2x)$, da graficarsi nell'intervallo $I = [0, 5]$

```
x=0:0.01:5;  
y1=x.*exp(-x);  
y2=x./(1+x)+sin(x).*cos(2*x);  
plot(x,y1,x,y2),grid
```



Per distinguere le due curve si può inserire una **legenda** (comando `legend`). Alla funzione `legend` devono essere passati come argomenti di ingresso un numero di **stringhe** pari al numero di grafici sovrapposti, in questo caso due. Nel codice di esempio che segue inseriamo anche una opzione di configurazione per mezzo della quale settare lo **spessore delle linee** del grafico.

```
x=0:0.01:5;  
y1=x.*exp(-x);  
y2=x./(1+x)+sin(x).*cos(2*x);  
plot(x,y1,x,y2,'LineWidth',2),grid  
legend('Curva y_1','Curva y_2')
```

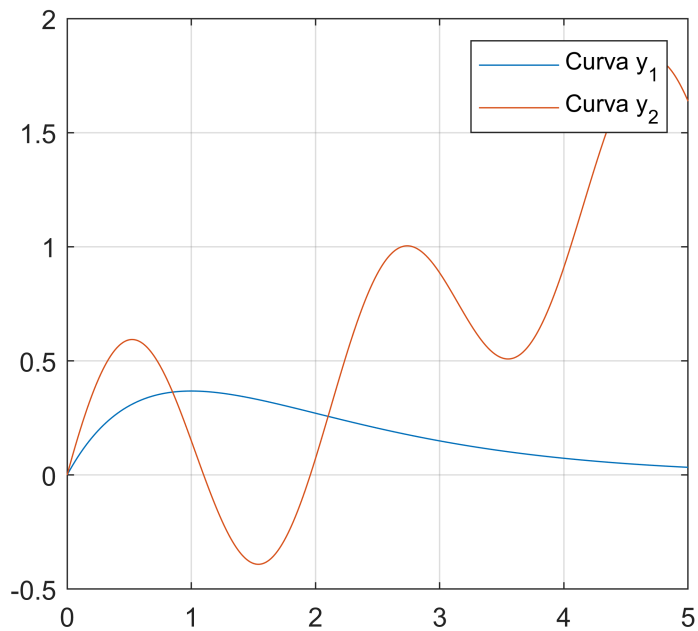



Per la creazione di grafici con curve sovrapposte si può utilizzare in alternativa una sintassi differente maggiormente compatta. Anziché passare in ingresso alla funzione plot le due (o più) coppie di vettori si utilizza la seguente sintassi:

`plot(x,M)`

in cui x è il vettore che contiene i punti dell'intervallo in cui si desidera tracciare il grafico (nel caso di segnali si tratterà degli istanti di campionamento) mentre M è una matrice le cui righe o colonne (in funzione del fatto che x sia un vettore riga o colonna) contengono i corrispondenti valori delle curve da graficare. La matrice M ed il vettore x devono avere dimensioni omogenee

```
x=0:0.01:5;
y1=x.*exp(-x);
y2=x./(1+x)+sin(x).*cos(2*x);
plot(x,[y1;y2]),grid
legend('Curva y_1','Curva y_2')
```

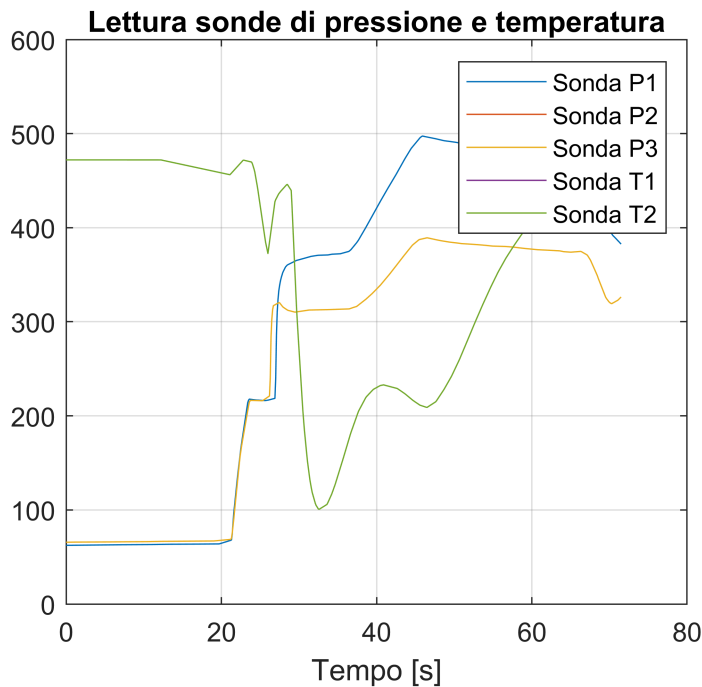


ESERCIZIO 1: Rappresentare in un unico grafico le misure dei 5 sensori contenute nella matrice M impiegata nel corso di un precedente esempio

SOLUZIONE

```
load M % carica il file M.mat
%estraiamo dalle colonne della matrice i vettori che contengono gli istanti
%di campionamento (prima colonna) e le letture dai 5 sensori (colonne dalla seconda alla sesta)
t=M(:,1);
T1=M(:,2);
P1=M(:,3);
P2=M(:,4);
T2=M(:,5);
P3=M(:,6);

plot(t,[P1 P2 P3 T1 T2]),grid
eval(s)
title('Lettura sonde di pressione e temperatura')
xlabel('Tempo [s]')
legend('Sonda P1','Sonda P2','Sonda P3','Sonda T1','Sonda T2')
```



ESERCIZIO 2

La funzione $y(x) = \frac{1}{(1-x)^2}$ ammette i seguenti sviluppi in serie di McLaurin di grado 2, 3 e 4

$$\hat{y}_2(x) = 1 + 2x + 3x^2 \quad \hat{y}_3(x) = 1 + 2x + 3x^2 + 4x^3 \quad \hat{y}_4(x) = 1 + 2x + 3x^2 + 4x^3 + 5x^4$$

Creare, mediante un file script, un grafico che riporti, sovrapposte, la funzione $y(x)$ ed i tre polinomi approssimanti sopra riportati nell'intervallo $x \in [0, 0.2]$. Il grafico dovrà essere corredato da adeguate etichette e legende esplicative che ne chiariscano il contenuto.

SOLUZIONE

```
x=0:0.001:0.2;
y=1./((1-x).^2);

y2=1+2*x+3*x.^2;
y3=1+2*x+3*x.^2+4*x.^3;
y4=1+2*x+3*x.^2+4*x.^3+5*x.^4;

plot(x,[y;y2;y3;y4]),grid
title('Funzione y(x)=1/(1-x)^2 e polinomi di McLaurin')
legend('Funzione','Pol. grado 2','Pol. grado 3','Pol. grado 4','Location','NorthWest')
```

Il codice soluzione descrive anche come scegliere la collocazione della legenda, in questo caso posizionata nella figura in alto a sinistra (NorthWest). Provare altre opzioni (ad es. West, SouthEast, etc)