
Matlab - Simulink

**Software per il calcolo e per
la simulazione della dinamica dei sistemi**

Il miglior modo per imparare un programma è quello di utilizzarlo.

- Per una panoramica sui comandi di Matlab digitare:
demo
e seguire le istruzioni.
- Per le spiegazioni su un comando noto digitare:
help nomecomando
- Per trovare un comando non noto riguardante un certo argomento digitare:
lookfor keyword

dove keyword è una parola relativa all'argomento di interesse.

Esempio:

```
lookfor transfer
```

restituisce vari comandi relativi alle funzioni di trasferimento

- Molti comandi specifici per i controlli automatici:
help control

I dati sperimentali e i risultati delle simulazioni sono spesso forniti in forma di vettori o matrici. Matlab gestisce i vettori e le matrici come descritto nelle demo:

demo -> Matrices -> ...

Un singolo elemento di una matrice A si ottiene con:

$A(n_riga, n_colonna)$

La colonna k -esima di una matrice A (è un vettore colonna) si ottiene con:

$A(:, k)$

La riga k -esima di una matrice A (è un vettore riga) si ottiene con:

$A(k, :)$

Un vettore riga (colonna) si trasforma in un vettore colonna (riga) con:

$V_riga = V_colonna'$

Matlab Command Window

I comandi in Matlab si inseriscono tramite il "Command Window".

Alcuni comandi di uso generale sono:

`pwd` restituisce la directory corrente

`dir` elenca i file della directory

`dir *.m` elenca i file .m

`cd newdir` cambia la directory di lavoro

`clc` pulisce la finestra

Le variabili definite in Matlab sono raccolte nel "WorkSpace".

`who` elenca le variabili del workspace

I file di comandi hanno estensione .m

e contengono dei comandi di Matlab


che sono eseguiti digitando il nome del file come se fosse un comando.

`motoreDC0` è un file comandi.

I file di comandi si scrivono con il

"Matlab Editor":

`edit motoreDC0`



```
MATLAB Command Window
File Edit View Window Help

This version is for educational classroom use only.

To get started, type one of these: helpwin, helpdesk, or demo.
For product information, type tour or visit www.mathworks.com.

>> pwd
ans =
C:\MATLABR11\work

>> dir
.          motoreDC0.m      motoreDC0plot.m
..         motoreDC0mdl.mdl

>> dir *.m
motoreDC0.m      motoreDC0plot.m

>> motoreDC0
elapsed_time =
    0.4060

>> who
Your variables are:

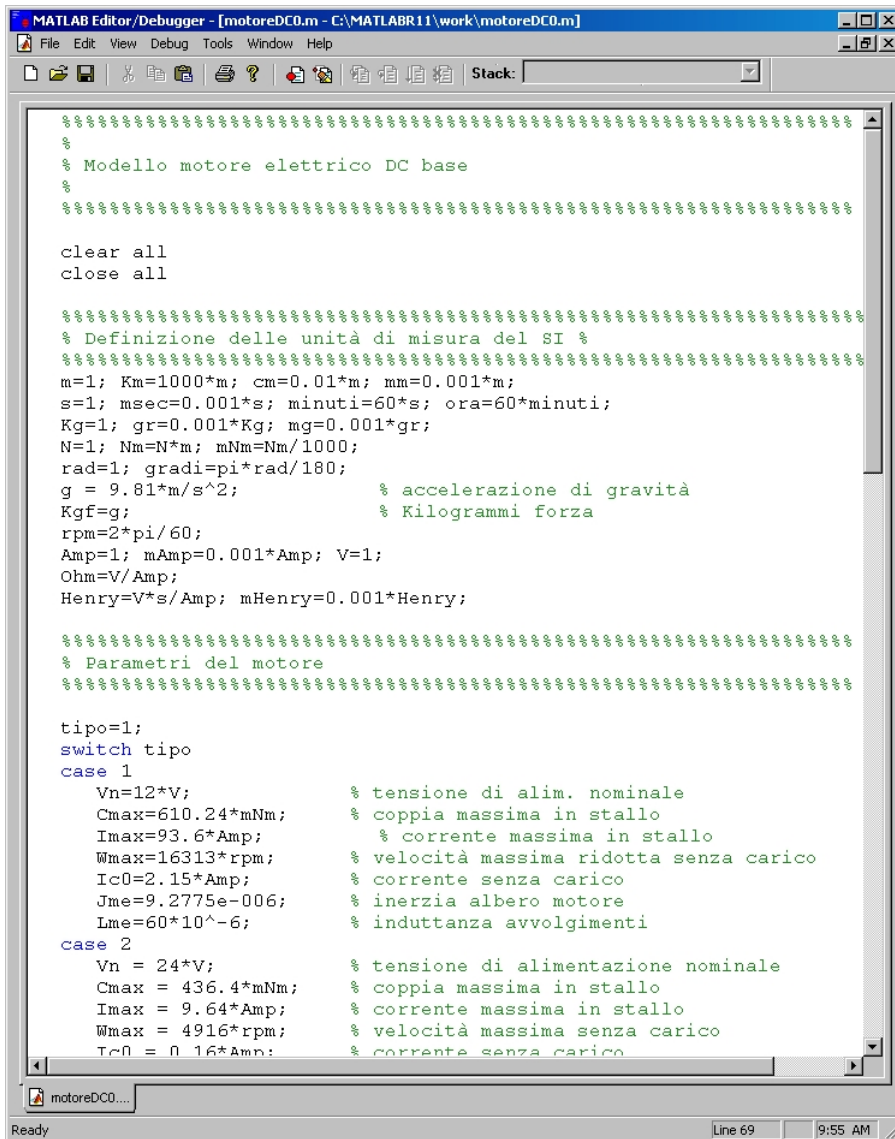
Amp      Kgf      U          h          rad
Cm       Km       Ucem       lw         rpm
Cmax     Lme      Vi         n           s
Cr       N        Un         mAmp       t
Henry    Nm       Wmax       mHenry     th
Ic0      Ohm      Wme        mNm        th0
Imax     Rme0     Wme0       mg         tipo
Ime      TABCR    bme        minuti     tout
Ime0     TABTEMPI cm         mm
Jme      TABVIN   g          msec
KI       Tcsim    gr         nprova
Kg       Tfin     gradi      ora

>> edit motoreDC0
>> cd d:
>> pwd
ans =
D:\

>> clc

Ready
```

Matlab Editor



The screenshot shows the MATLAB Editor/Debugger interface. The title bar reads 'MATLAB Editor/Debugger - [motoreDC0.m - C:\MATLABR11\work\motoreDC0.m]'. The menu bar includes File, Edit, View, Debug, Tools, Window, and Help. The toolbar contains icons for file operations and debugging. The main editor window displays a script for a DC motor model. The script includes comments in Italian, variable definitions for SI units, and a switch statement for motor parameters. The status bar at the bottom indicates 'Ready', 'Line 69', and '9:55 AM'.

```
#####  
%  
% Modello motore elettrico DC base  
#####  
  
clear all  
close all  
  
#####  
% Definizione delle unità di misura del SI %  
#####  
m=1; Km=1000*m; cm=0.01*m; mm=0.001*m;  
s=1; msec=0.001*s; minuti=60*s; ora=60*minuti;  
Kg=1; gr=0.001*Kg; mg=0.001*gr;  
N=1; Nm=N*m; mNm=Nm/1000;  
rad=1; gradi=pi*rad/180;  
g = 9.81*m/s^2; % accelerazione di gravità  
Kgf=g; % Kilogrammi forza  
rpm=2*pi/60;  
Amp=1; mAmp=0.001*Amp; V=1;  
Ohm=V/Amp;  
Henry=V*s/Amp; mHenry=0.001*Henry;  
  
#####  
% Parametri del motore  
#####  
  
tipo=1;  
switch tipo  
case 1  
    Vn=12*V; % tensione di alim. nominale  
    Cmax=610.24*mNm; % coppia massima in stallo  
    Imax=93.6*Amp; % corrente massima in stallo  
    Wmax=16313*rpm; % velocità massima ridotta senza carico  
    Ic0=2.15*Amp; % corrente senza carico  
    Jme=9.2775e-006; % inerzia albero motore  
    Lme=60*10^-6; % induttanza avvolgimenti  
case 2  
    Vn = 24*V; % tensione di alimentazione nominale  
    Cmax = 436.4*mNm; % coppia massima in stallo  
    Imax = 9.64*Amp; % corrente massima in stallo  
    Wmax = 4916*rpm; % velocità massima senza carico  
    Ic0 = 0.16*Amp; % corrente senza carico
```

L'editor di Matlab permette di scrivere i file di comandi (con estensione .m) che contengono una successione di comandi che saranno eseguiti da Matlab quando si digita il nome del file (senza .m) nel command window.

I file comandi sono molto utili per facilitare tutte le operazioni con Matlab. In particolare è possibile ripetere lunghe sequenze di comandi senza doverli riscrivere direttamente nel Command Window.

La possibilità di modificare facilmente il valore delle variabili permette di effettuare velocemente calcoli, simulazioni e confronti.

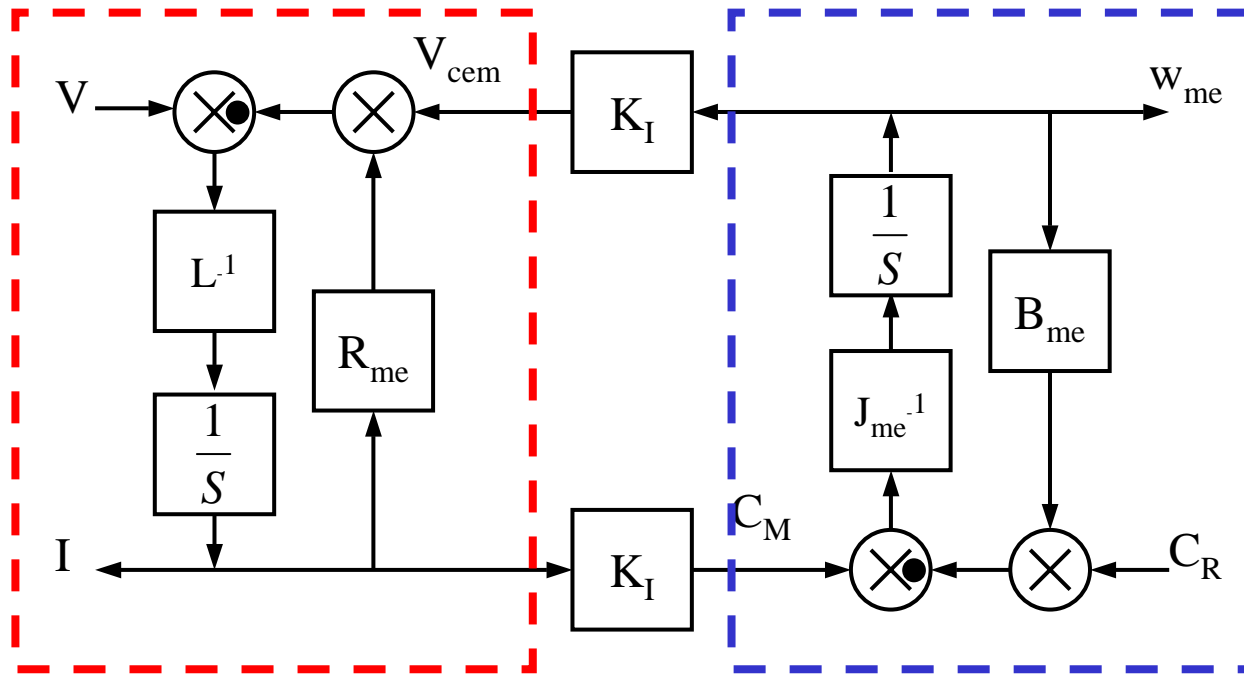
I commenti sono preceduti da %
% questo è un commento

I modelli realizzati con Matlab-Simulink sono strutturati solitamente in tre file principali:

- Modello Simulink (`motoreDC0mdl.mdl`)
Contiene il modello del sistema da simulare costruito con i blocchi di Simulink. Memorizza i segnali della simulazione.
- File (di comandi) dei parametri del modello (`motoreDC0.m`).
È il file principale per chiamare la simulazione. Contiene:
 - Conversioni delle unità di misura
 - Parametri del modello
 - Condizioni iniziali della simulazione
 - Segnali di ingresso del sistema
 - Chiamata al file di graficazione dei risultati
- File (di comandi) per la graficazione risultati (`motoreDC0plot.m`)
Permette di rappresentare e di elaborare i segnali rilevati durante la simulazione del modello Simulink.

NOTA BENE: per chiarezza e comodità, è meglio evitare di inserire valori numerici direttamente nello schema Simulink, sebbene sia possibile.

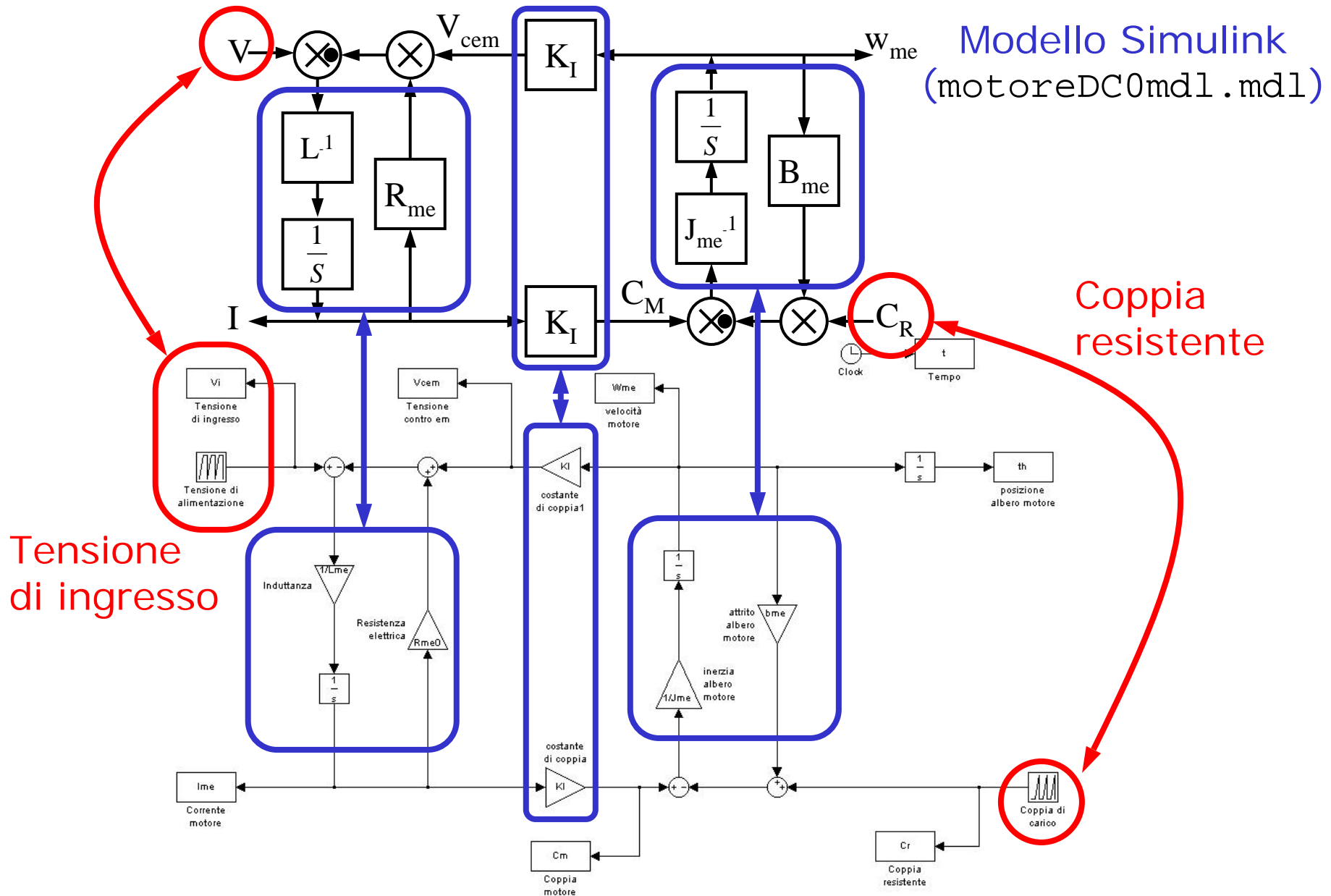
Modello POG Motore in Corrente Continua



Dominio
Elettrico

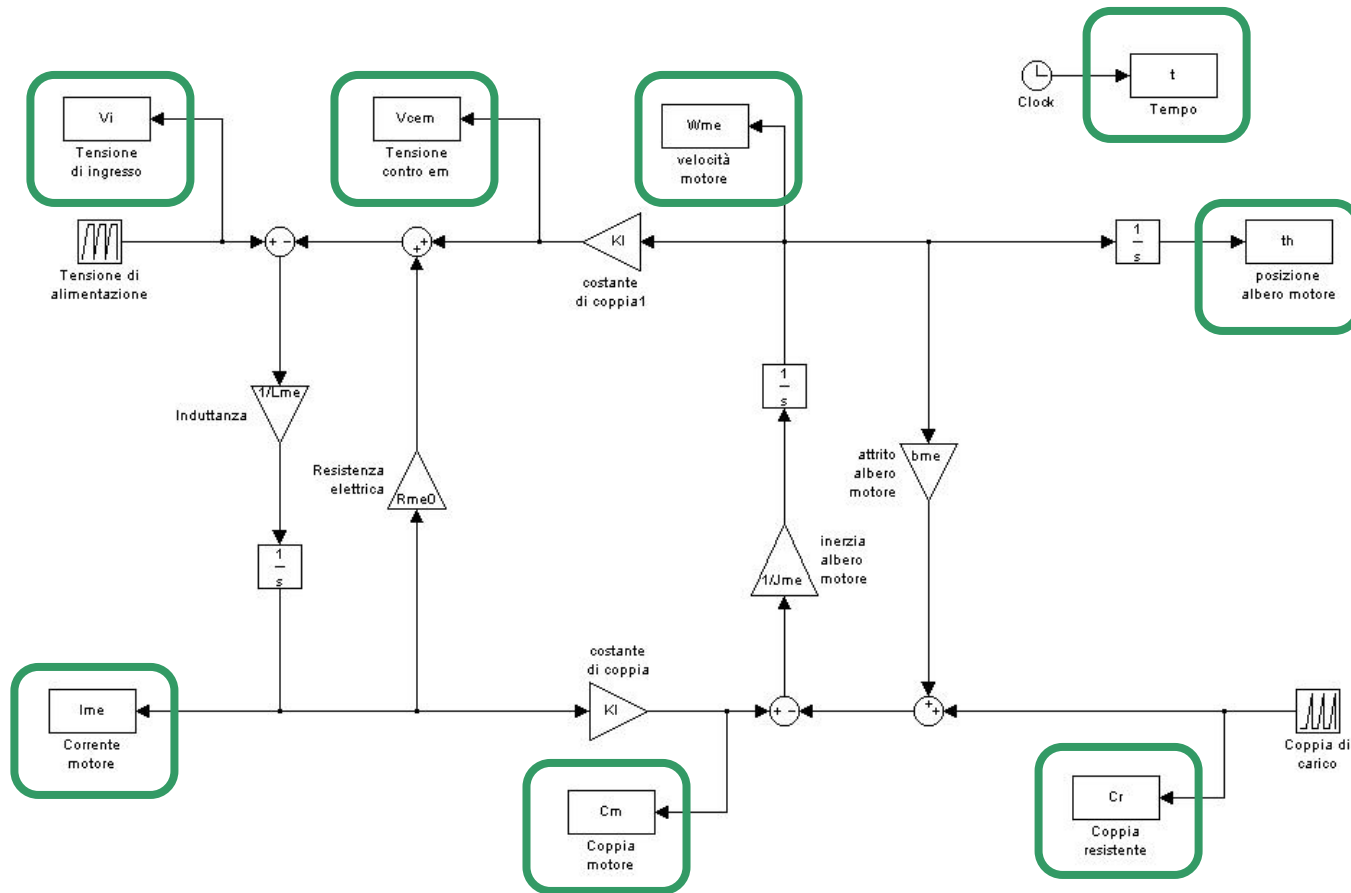
Dominio
Meccanico

Corrispondenza Modello POG e Modello Simulink



Memorizzazione dei segnali di una simulazione

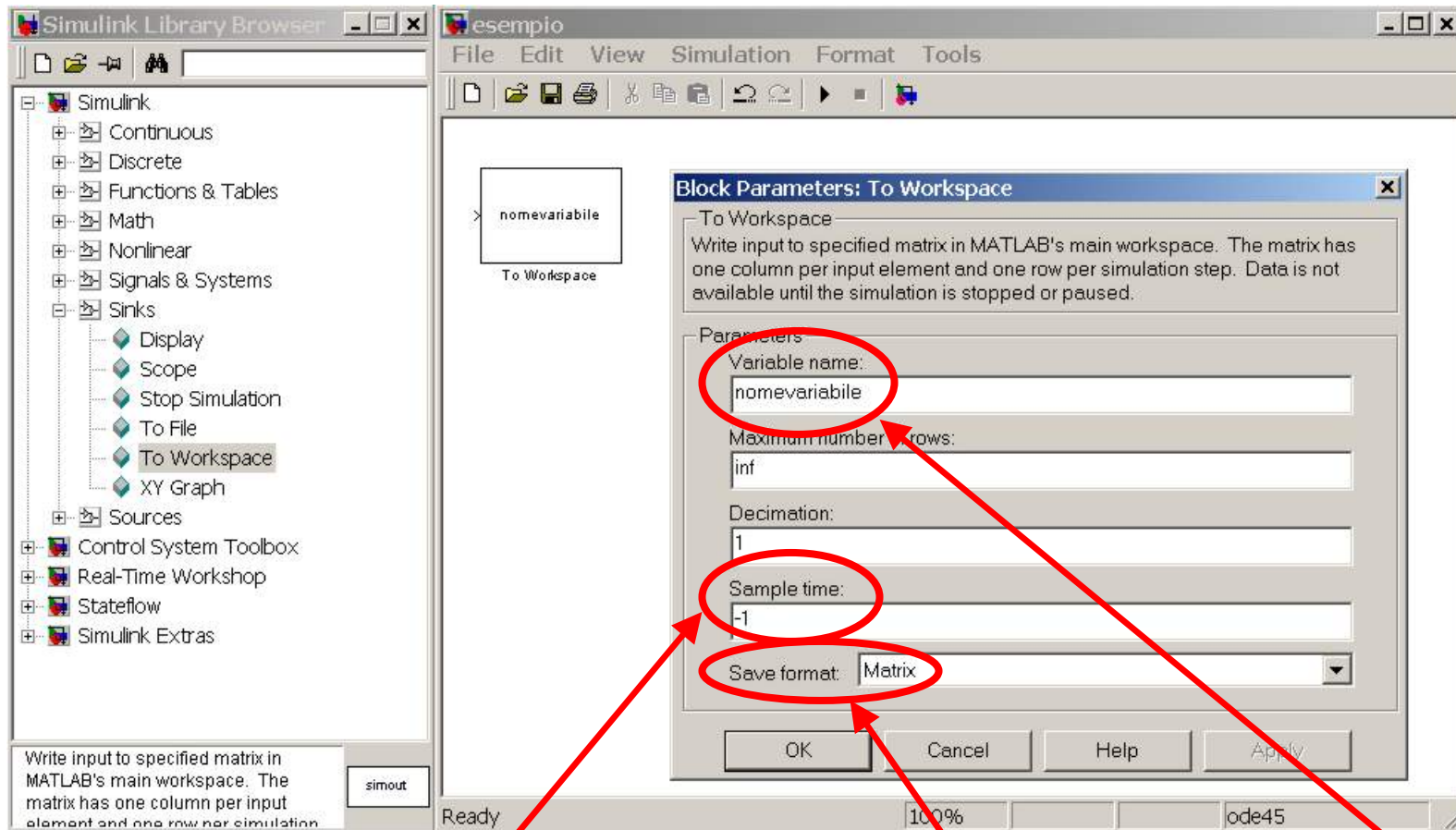
I segnali simulati si possono “misurare” e memorizzare in variabili di tipo array che saranno utilizzate per rappresentare i risultati.



Per memorizzare i segnali di una simulazione si usa il blocco Sinks -> To Workspace che permette di memorizzare i segnali come variabili nel workspace

Memorizzazione dei segnali di una simulazione

Il blocco To Workspace permette di definire:



Tempo di campionamento dei dati della simulazione:

-1 = definito da Simulink

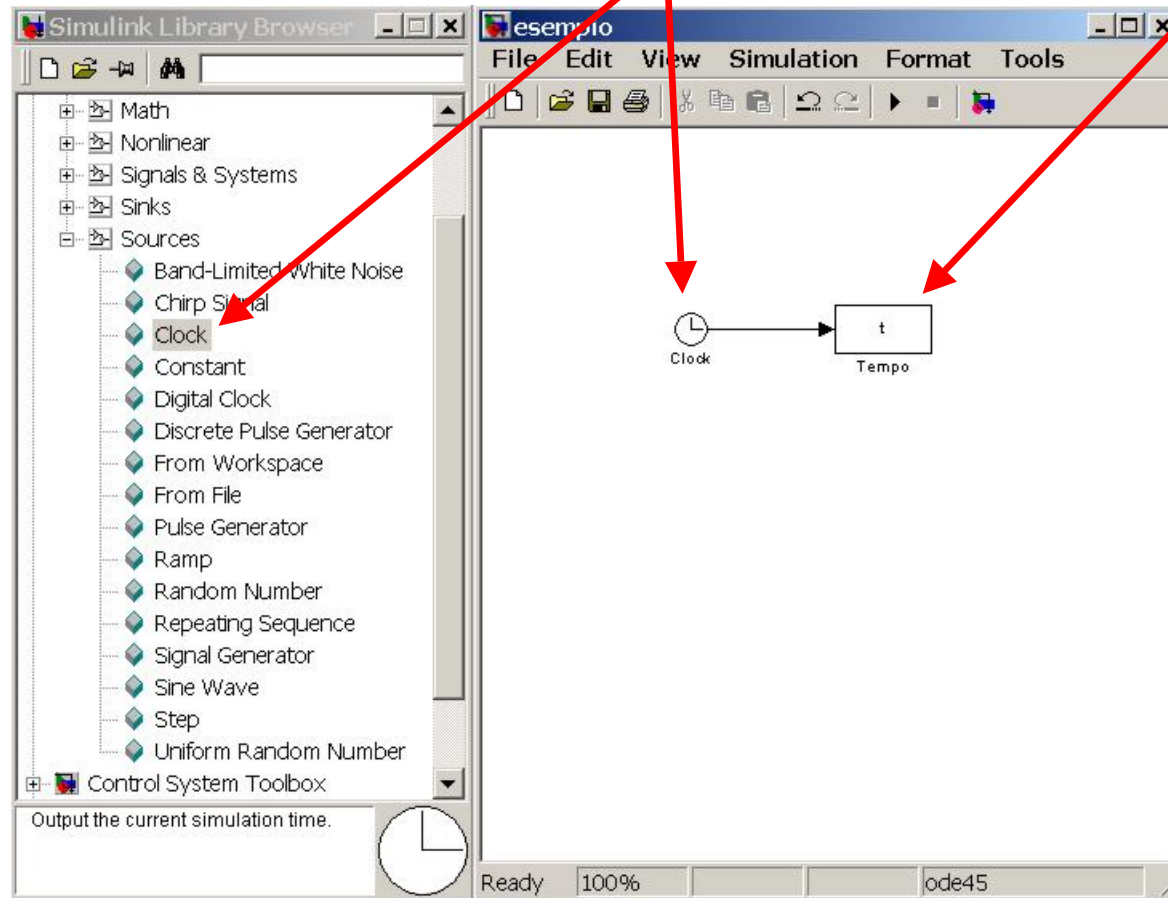
Tcsim = definito dall'utente

Dati salvati come vettori o matrici

Nome della variabile

Memorizzazione dei segnali di una simulazione

Gli istanti di tempo che corrispondono ai dati memorizzati si ottengono usando la sorgente "Clock" e il blocco "ToWorkspace"



La variabile **t** è un vettore colonna con n righe.

Tutti i dati memorizzati tramite i blocchi "ToWorkspace" hanno n righe e un numero di colonne pari alla dimensione della variabile.

Memorizzazione dei segnali di una simulazione

Si possono imporre gli istanti di tempo in cui memorizzare i segnali nei blocchi "ToWorkspace" ponendo la variabile "sample time" al valore desiderato (Es: 0.001 ovvero 1ms).

In alternativa si può definire prima della simulazione il vettore dei tempi:

```
% simulazione  
Tsim=[0:0.001:5];  
  
sim('motoreDC0mdl',Tsim)
```

In questo esempio il vettore del tempo avrà valori 0, 0.001, 0.002, ... 4.999, 5.000 e tutte le variabili del modello Simulink saranno "campionate" in corrispondenza di tali istanti.

La struttura consigliata per i file dei parametri è la seguente:

1. definizione delle unità di misura e delle conversioni
2. Parametri del modello
3. Caricamento dei dati sperimentali
4. Condizioni iniziali della simulazione
5. Segnali di ingresso della simulazione
6. Parametri della Simulazione e Simulazione
7. Graficazione dei risultati della simulazione

È tipicamente conveniente (specialmente quando le simulazioni richiedono molto tempo) gestire separatamente un file di comandi per la graficazione dei risultati delle simulazioni (motoreDC0plot.m)

File dei Parametri – Definizione delle unità di misura e conversioni

Le unità di misura consigliate sono quelle del sistema internazionale.
Per facilitare le conversioni nel sistema internazionale i fattori di conversione sono riportati all'inizio del file dei parametri motoreDC0.m

```
% Definizione delle unità di misura del SI
m=1; Km=1000*m; cm=0.01*m; mm=0.001*m;
s=1; msec=0.001*s; minuti=60*s; ora=60*minuti;
Kg=1; gr=0.001*Kg; mg=0.001*gr;
N=1; Nm=N*m; mNm=Nm/1000;
rad=1; gradi=pi*rad/180;
g = 9.81*m/s^2;           % accelerazione di gravità
Kgf=g;                   % Kilogrammi forza
rpm=2*pi/60;
Amp=1; mAmp=0.001*Amp; V=1;
Ohm=V/Amp;
Henry=V*s/Amp; mHenry=0.001*Henry;
```

Esempio: scrivendo

```
Vmax = 10000*rpm;
```

si ottiene V_{\max} con valore in [rad/s] (unità SI)
pur avendolo scritto in [rpm]

File dei Parametri – Parametri del modello

Tutti i parametri del modello Simulink sono contenuti nel file dei parametri (motoreDC0.m)

```
% Parametri del motore
```

```
tipo=1;
```

```
switch tipo
```

```
case 1
```

```
Vn=12*V;
```

```
Cmax=610.24*mNm;
```

```
Imax=93.6*Amp;
```

```
Wmax=16313*rpm;
```

```
Ic0=2.15*Amp;
```

```
Lme=60*10^-6;
```

```
Jme=9.2775e-006;
```

```
case 2
```

```
...
```

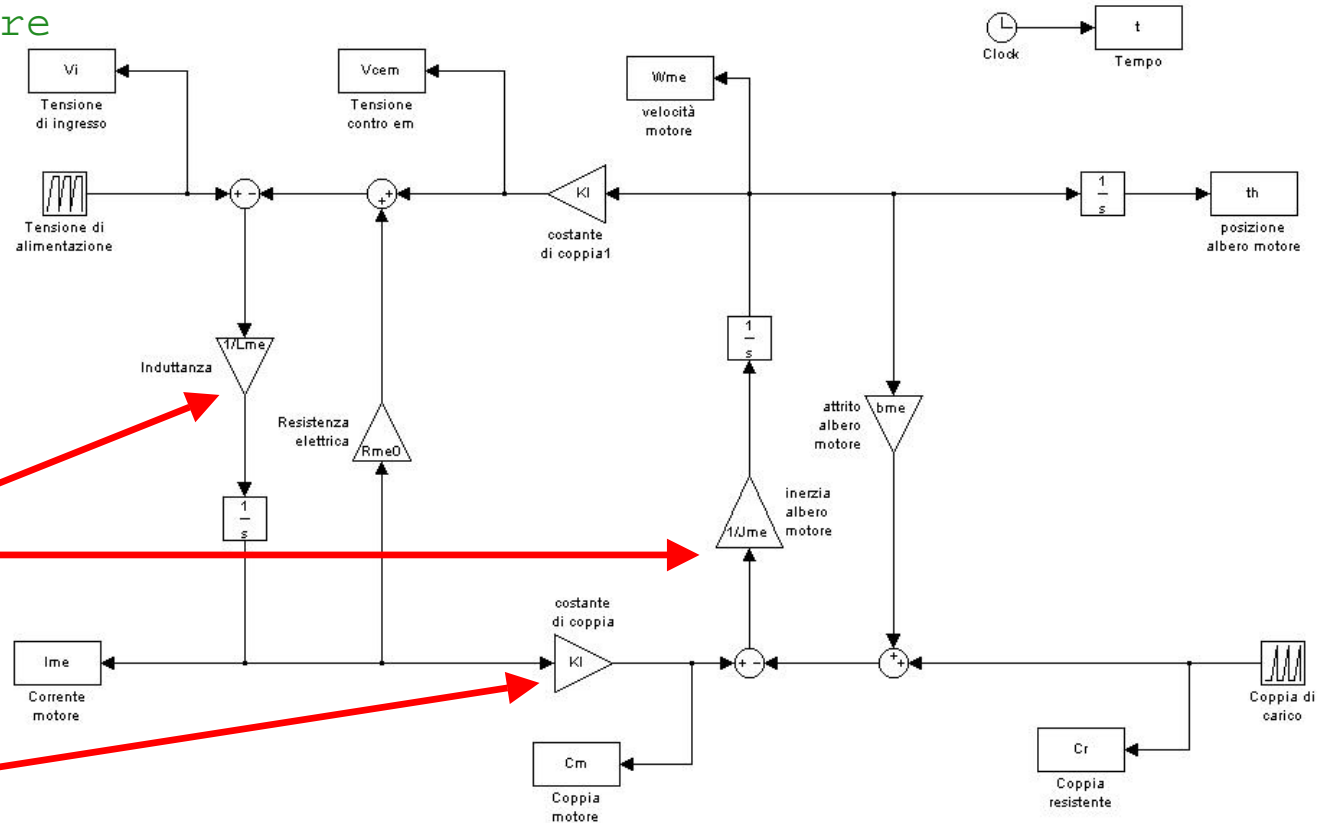
```
end
```

```
% calcolo parametri  
del modello
```

```
KI=Cmax/Imax;
```

```
Rme0=Vn/Imax;
```

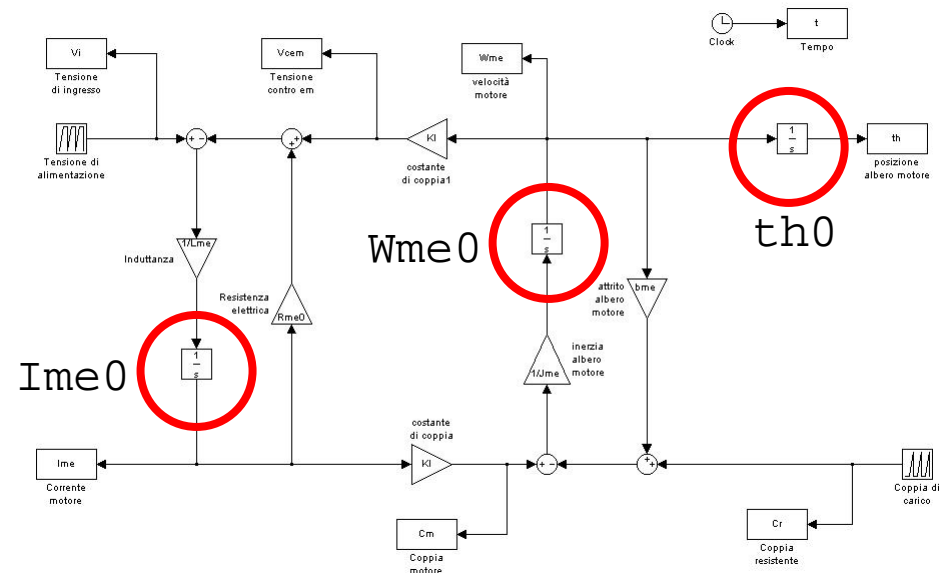
```
bme=Ic0*KI/Wmax;
```



File dei Parametri – Condizioni iniziali

Ad ogni **integratore** del modello corrisponde una **condizione iniziale**
Tutte le condizioni iniziali sono contenute nel file motoreDC0.m

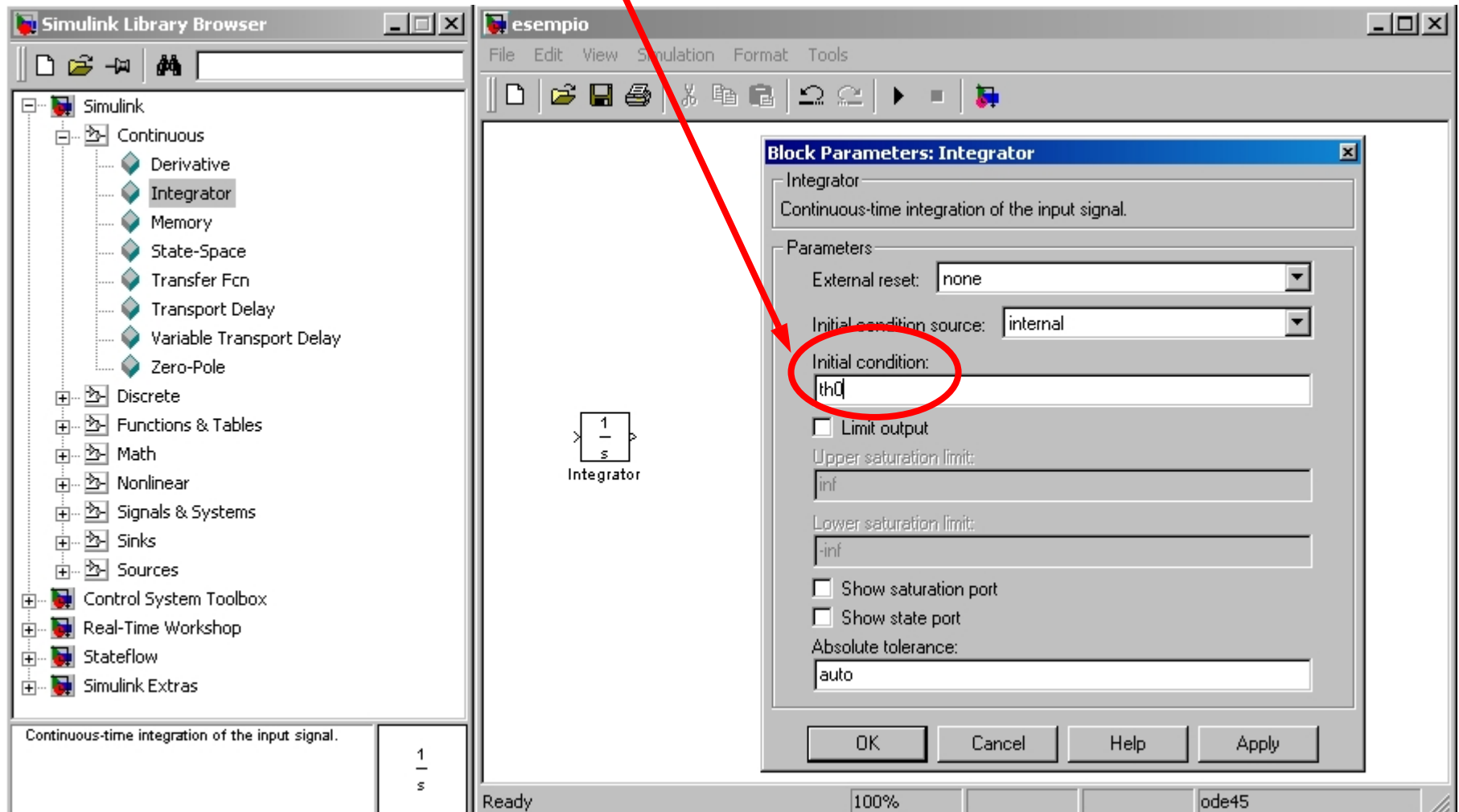
```
% dati di ingresso e condizioni iniziali
nprova=1;
switch nprova
case 1
    % Rilievo caratteristica statiche
    th0=0 % posizione iniziale albero
    Ime0=Imax; % corrente iniziale
    Wme0=0; % velocità iniziale motore
    Tfin=60; % tempo della simulazione
    TABTEMPI=[0 Tfin/2.2 Tfin/1.8 Tfin];
    TABCR=[Cmax 0 0 Cmax];
    TABVIN=[Vn Vn Vn Vn];
case 2
    ...
end
```



File dei Parametri – Condizioni iniziali

Ad ogni **integratore** del modello corrisponde una **condizione iniziale**
Tutte le condizioni iniziali sono contenute nel file motoreDC0.m

```
th0=0 % posizione iniziale albero
```



File dei Parametri – Segnali di ingresso

I segnali di ingresso sono contenuti nel file motoreDC0.m

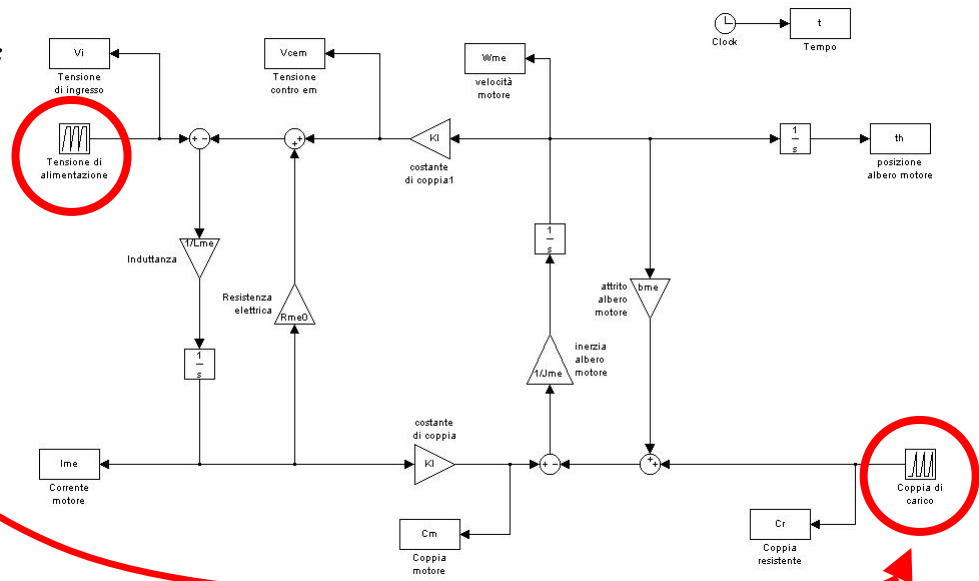
```
% dati di ingresso e condizioni iniziali
nprova=2;
switch nprova
case 1
    % Rilievo caratteristiche statiche del
    motore
    th0=0;                % posizione
    iniziale albero motore
    Ime0=Imax;            % corrente iniziale sul
    motore
    Wme0=0;               % velocità iniziale motore
    Tfin=60;               % tempo della
    simulazione
```

```
TABTEMPI=[0 Tfin/2.2 Tfin/1.8 Tfin];
TABVIN=[Vn Vn Vn Vn];
TABCR=[Cmax 0 0 Cmax];
```

```
case 2
```

```
...
```

```
end
```

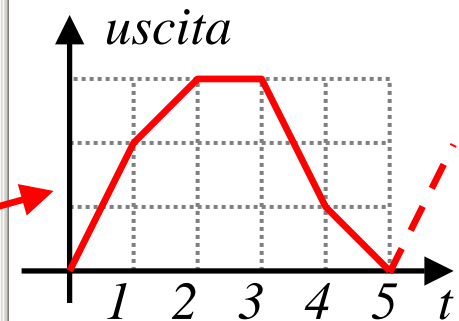
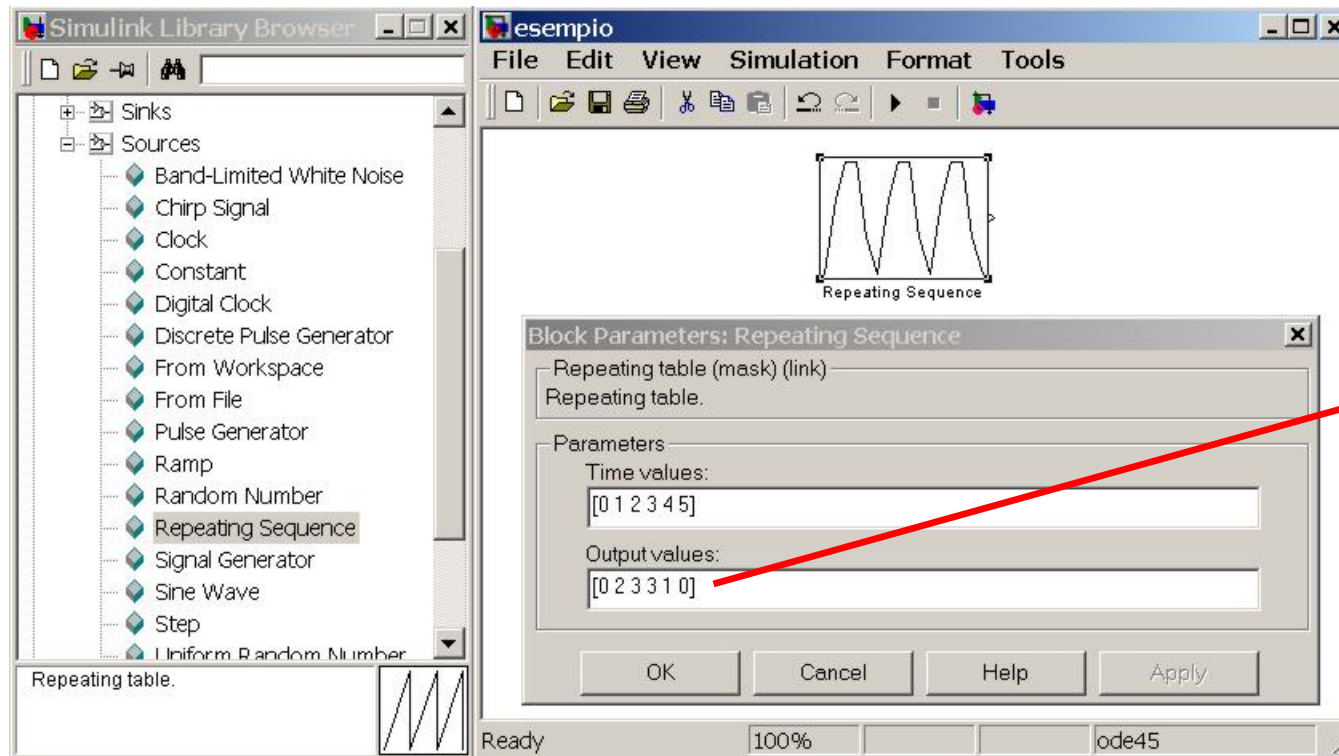


Ci sono diversi modi per inserire segnali di ingresso al sistema. Le sorgenti più utilizzate sono contenute nella libreria "Sources"

È opportuno che anche i segnali di ingresso siano parametrizzati quindi nel file Simulink devono comparire delle variabili il cui valore è definito nel file dei parametri (`motoreDC0.m`).

Segnali di ingresso

Molto utili sono le "Repeating sequence" in cui si può dare una serie di valori della variabile di uscita in corrispondenza di determinati istanti. Le coppie istante-valore sono poi interpolate linearmente e ripetute nel tempo. È opportuno parametrizzare nel file dei parametri i vettori "Time values" e "Output values".



Esempio: coppia di
carico al motore.

```
% dati di ingresso e condizioni iniziali  
...  
TABTEMPI=[0 Tfin/2.2 Tfin/1.8 Tfin];  
TABCR=[Cmax 0 0 Cmax];
```

File dei Parametri – Simulazione

Dopo il “caricamento” di tutti i dati necessari il file dei parametri “chiama” la simulazione del modello.

```
% simulazione
```

```
tic  
sim('motoreDC0mdl',Tfin)  
toc
```

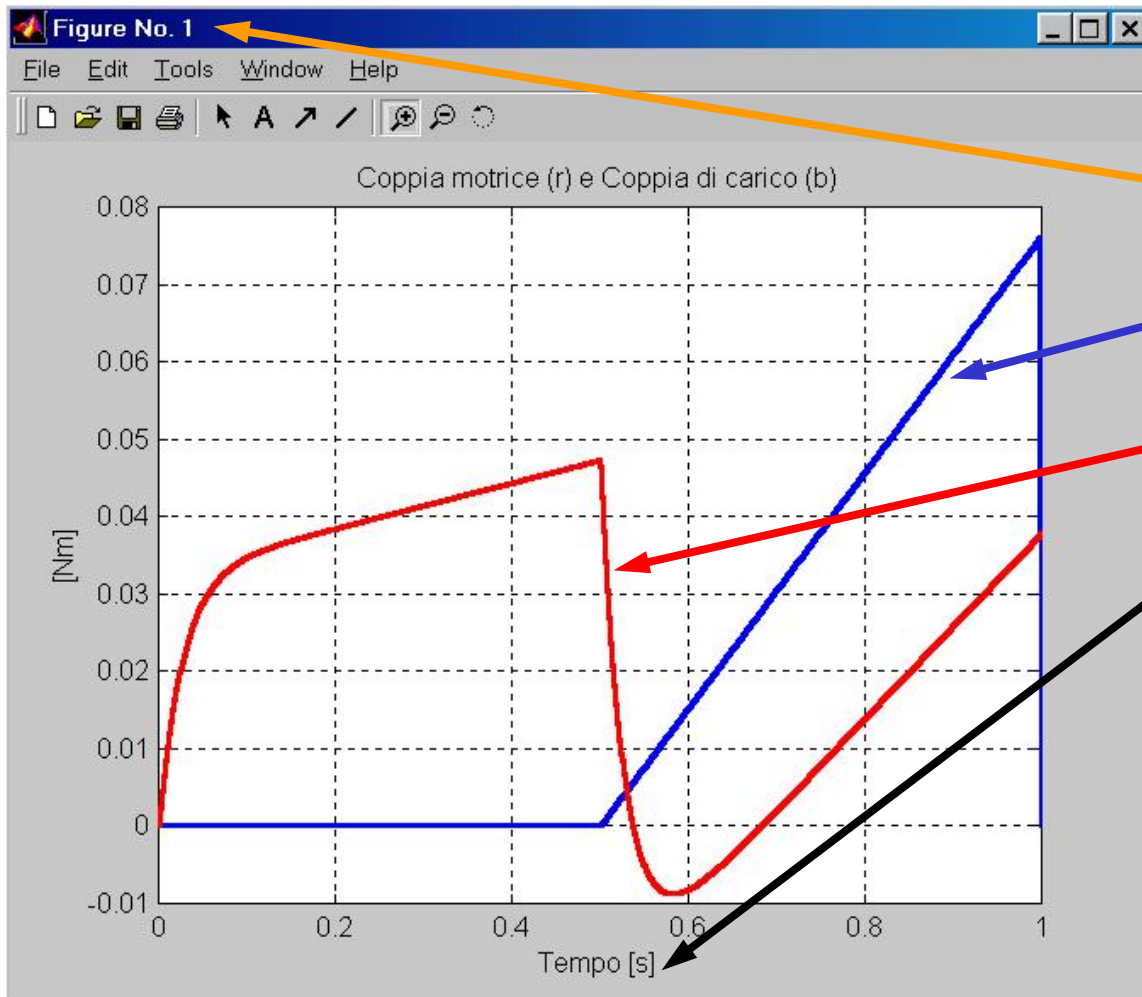
Misura del tempo
di simulazione

Istante finale
simulazione

```
% graficazione dei risultati  
motoreDC0plot
```

Al termine della simulazione il file dei parametri “chiama”
il file matlab `motoreDC0plot.m` per la graficazione dei risultati.

Tutti i segnali rilevati nella simulazione possono essere riportati in grafici. Il file `motoreDC0plot.m` contiene i comandi per la rappresentazione delle figure di interesse.



```
figure(1)
clf
plot (t,Cr,'b')
hold on
plot (t,Cm,'r')
title ('Coppia
motrice ...')
xlabel ('Tempo [s]')
ylabel ('[Nm]')
grid on
zoom on
```

I dati sperimentali devono essere organizzati in una matrice in cui ogni colonna rappresenta un segnale misurato. Tipicamente la prima colonna rappresenta il tempo.

Utilizzando l'editor di matlab (comando `edit`) si crea un file `.m` (esempio `misura1.m`) che carica i dati sperimentali nel workspace. Il contenuto tipico di questi file è il seguente:

```
% misura caratteristica statica...  
  
dati=[0      1      0.2  
      0.01    1.1    0.25  
      0.02    1.2    0.35  
      ...     ...     ...  
      5.46    7.8    0.96];  
  
TEMPO = dati(:,1);  
POSIZIONE = dati(:,2)*0.01;  
...
```

} Descrizione della misura
e dei dati rilevati

} Matrice dei
dati misurati

} Assegnamento dei
dati in vettori colonna
con conversione delle
unità di misura

Digitando dal workspace o all'interno di un file .m il nome del file
`misura1`

nel workspace vengono caricate le variabili definite nel file
`misura1.m` quindi: `dati`, `TEMPO`, `POSIZIONE`,....

Per distinguere facilmente le variabili corrispondenti a dati
sperimentali da quelle delle simulazioni è utile seguire la
convenzione di scrivere in caratteri tutti maiuscoli le variabili
sperimentali, esempio `TEMPO`, `POSIZIONE`,....

Per inserire direttamente i dati sperimentali come sorgenti di segnale negli schemi Simulink si utilizza il blocco:

Sources -> From workspace
nel seguente modo:

