

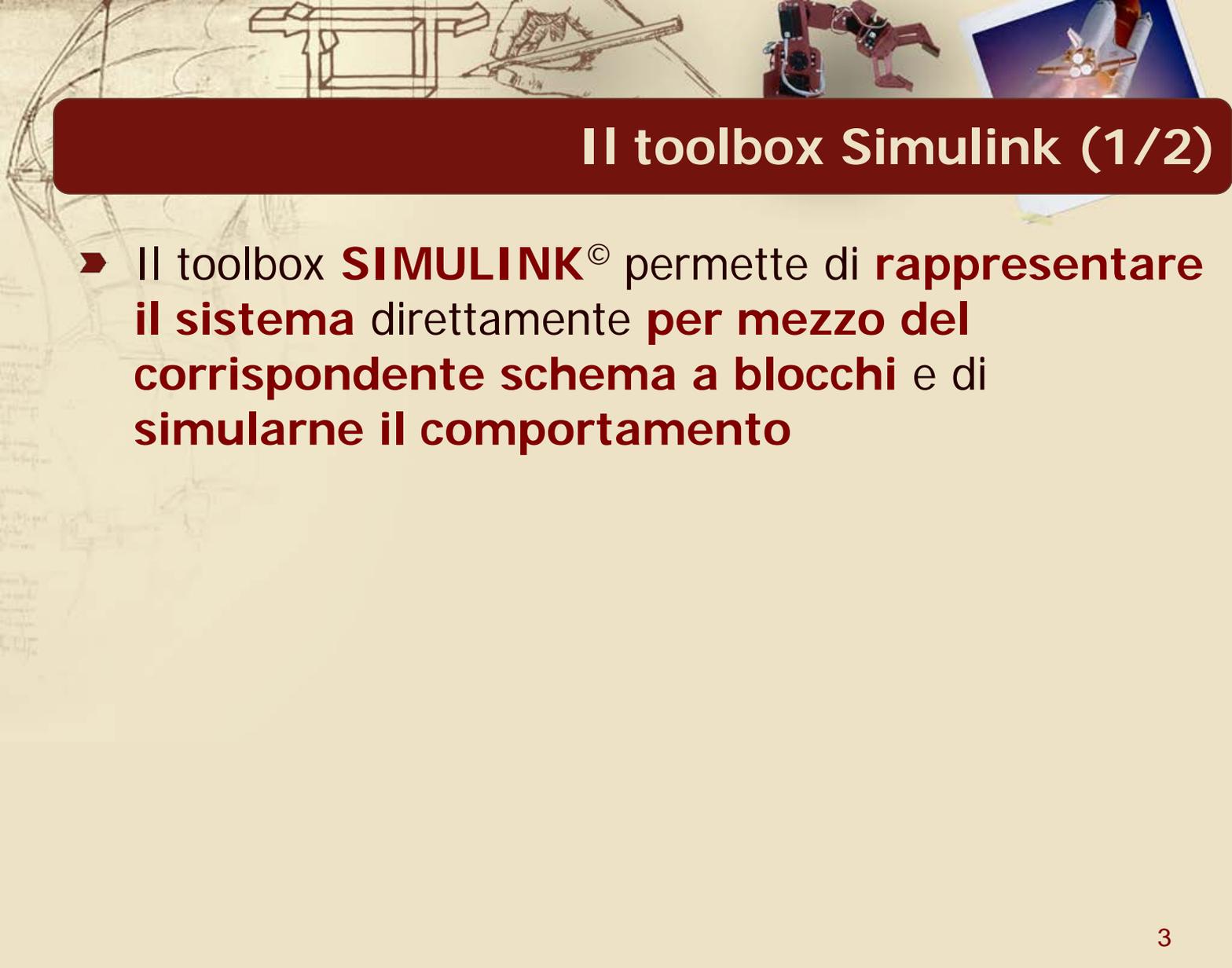
**Matlab e Simulink per i sistemi di controllo**

**Introduzione all'utilizzo di Simulink**



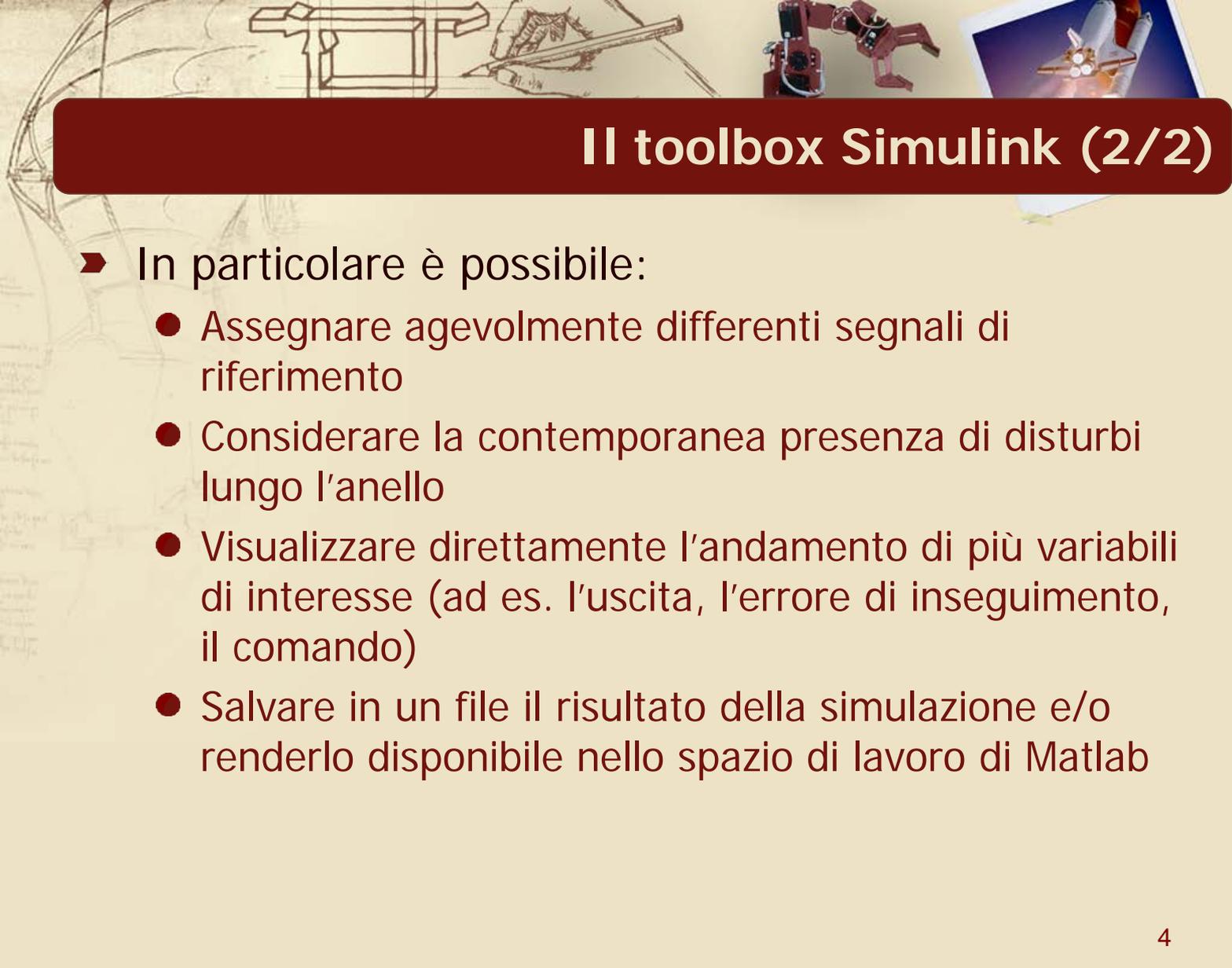
# Introduzione

- L'utilizzo del toolbox **SIMULINK**® di Matlab facilita la simulazione di sistemi interconnessi, consentendo la loro rappresentazione direttamente per mezzo del corrispondente schema a blocchi
- L'utilizzo congiunto di Matlab e Simulink permette di sviluppare interamente il progetto di un sistema di controllo, verificando agevolmente il soddisfacimento delle specifiche di progetto, nonché la valutazione delle prestazioni di interesse



## Il toolbox Simulink (1/2)

- Il toolbox **SIMULINK**® permette di **rappresentare il sistema** direttamente **per mezzo del corrispondente schema a blocchi** e di **simularne il comportamento**

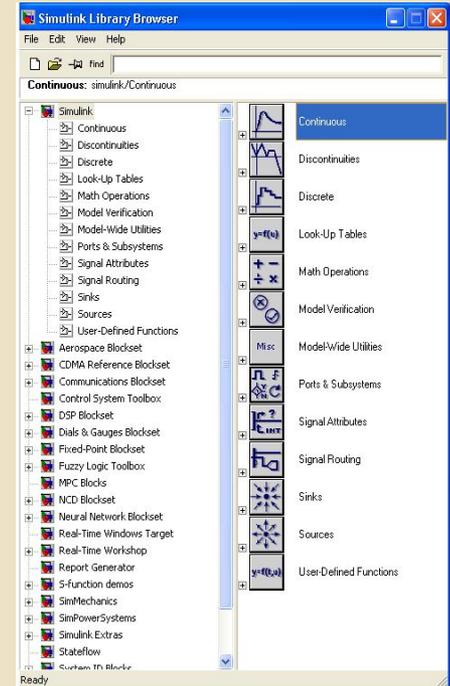


## Il toolbox Simulink (2/2)

- In particolare è possibile:
  - Assegnare agevolmente differenti segnali di riferimento
  - Considerare la contemporanea presenza di disturbi lungo l'anello
  - Visualizzare direttamente l'andamento di più variabili di interesse (ad es. l'uscita, l'errore di inseguimento, il comando)
  - Salvare in un file il risultato della simulazione e/o renderlo disponibile nello spazio di lavoro di Matlab

# Apertura di Simulink (1/2)

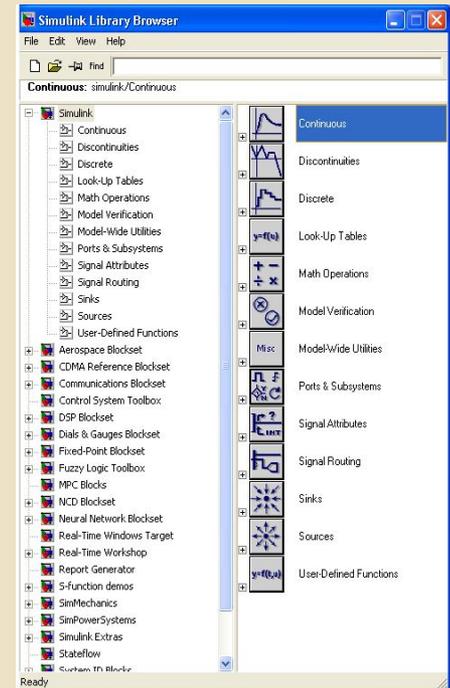
- Per **aprire Simulink**, è sufficiente digitare la parola "simulink" nella finestra di comando di Matlab oppure cliccare sulla corrispondente icona 

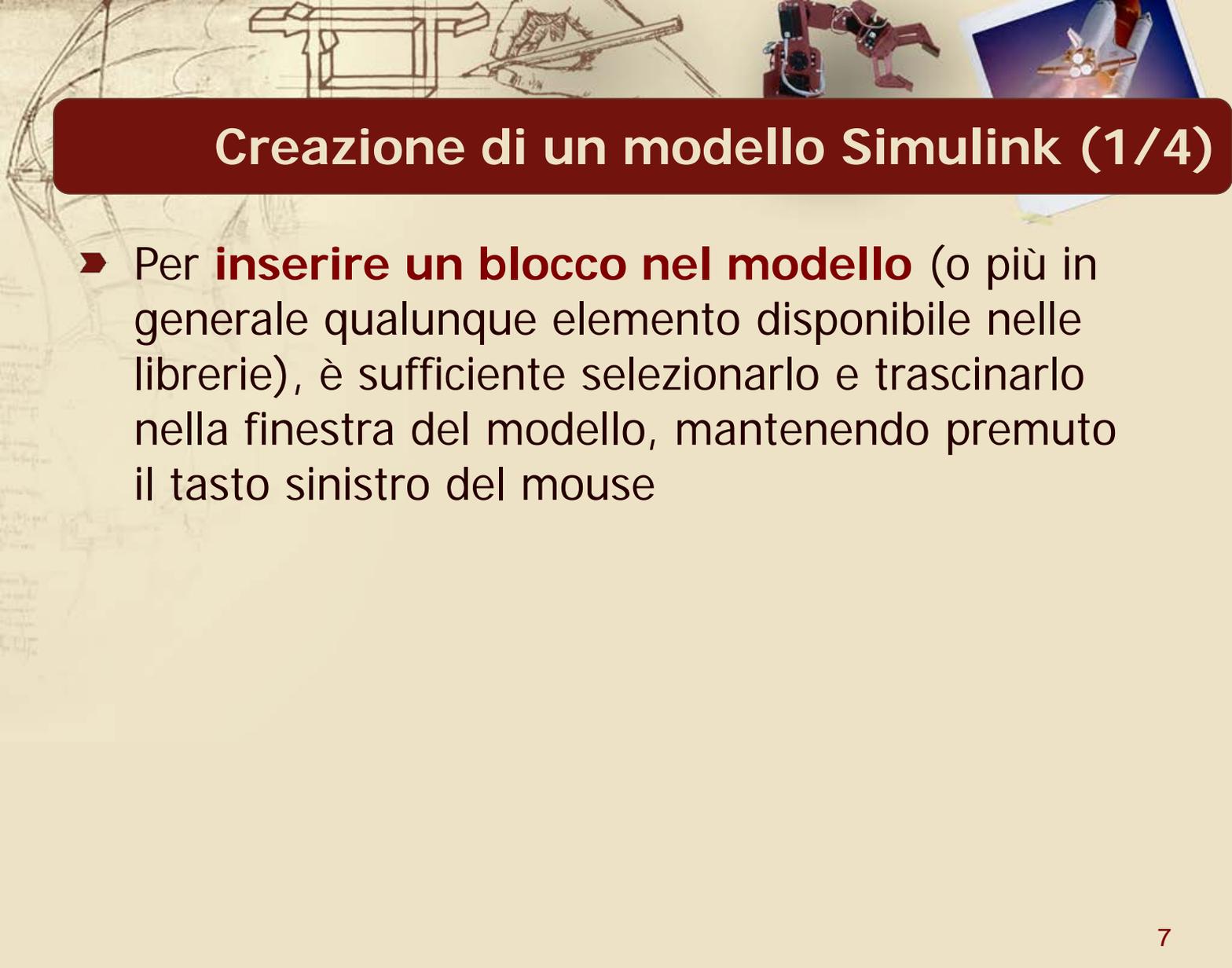


# Apertura di Simulink (2/2)

► Dalla finestra del “Simulink Library Browser” è possibile:

- Creare un nuovo modello o aprirne uno già esistente (file .mdl) dal menu “File”
- Individuare gli elementi di interesse nella libreria principale di Simulink o fra i toolbox disponibili
- Modificare alcune proprietà generali (ad esempio le caratteristiche dei font utilizzati)



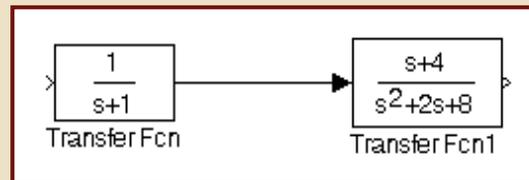


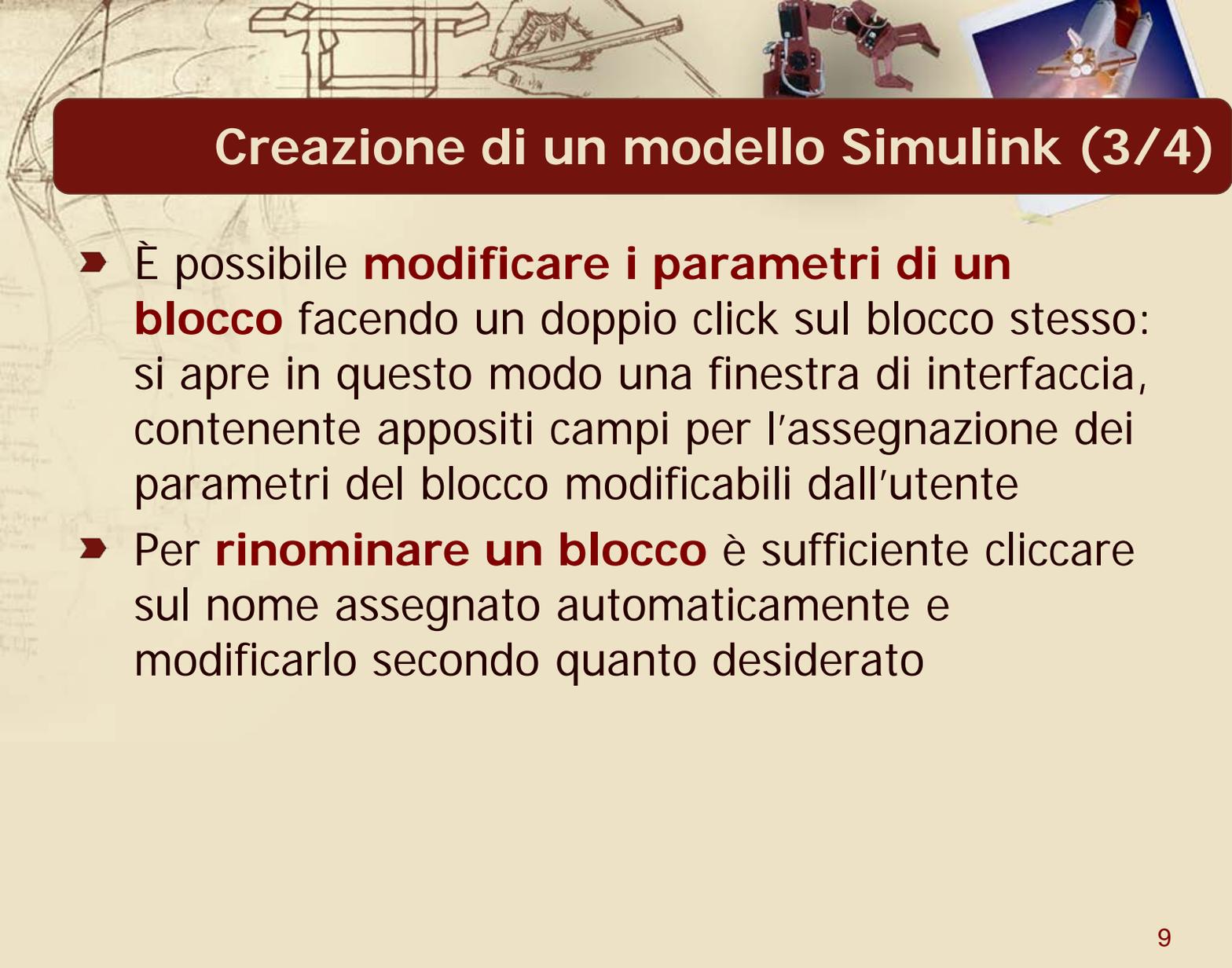
## Creazione di un modello Simulink (1/4)

- Per **inserire un blocco nel modello** (o più in generale qualunque elemento disponibile nelle librerie), è sufficiente selezionarlo e trascinarlo nella finestra del modello, mantenendo premuto il tasto sinistro del mouse

## Creazione di un modello Simulink (2/4)

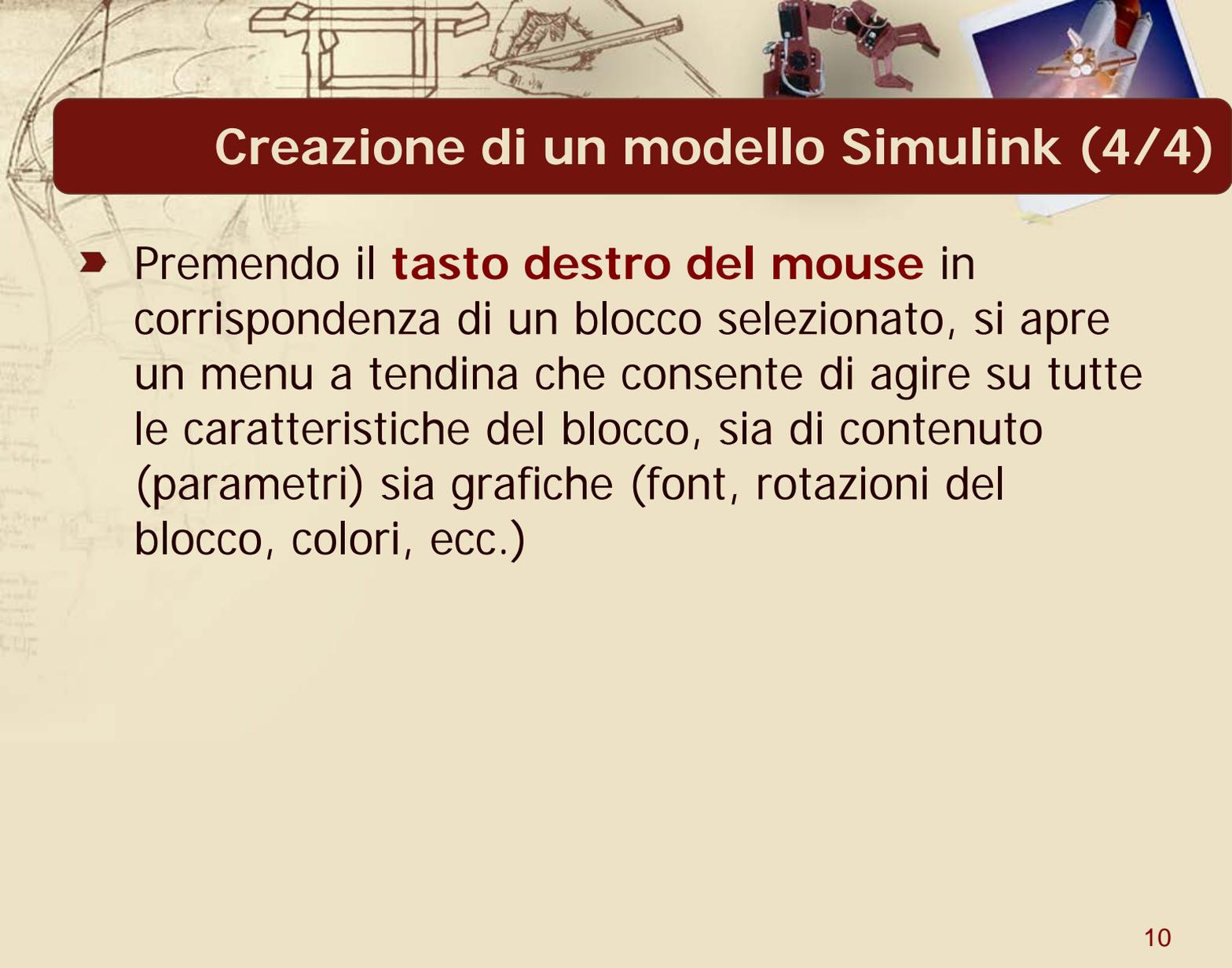
- Per **collegare due blocchi mediante un ramo**, è sufficiente cliccare sul morsetto di uscita del primo e trascinare il cursore (tenendo premuto il tasto sinistro del mouse) fino a raggiungere l'ingresso del secondo oppure selezionare i due blocchi (nella sequenza desiderata) tenendo premuto il tasto CTRL





## Creazione di un modello Simulink (3/4)

- È possibile **modificare i parametri di un blocco** facendo un doppio click sul blocco stesso: si apre in questo modo una finestra di interfaccia, contenente appositi campi per l'assegnazione dei parametri del blocco modificabili dall'utente
- Per **rinominare un blocco** è sufficiente cliccare sul nome assegnato automaticamente e modificarlo secondo quanto desiderato

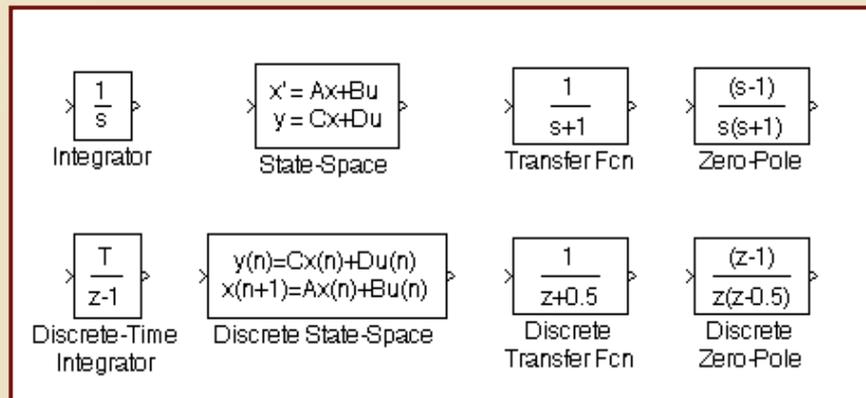


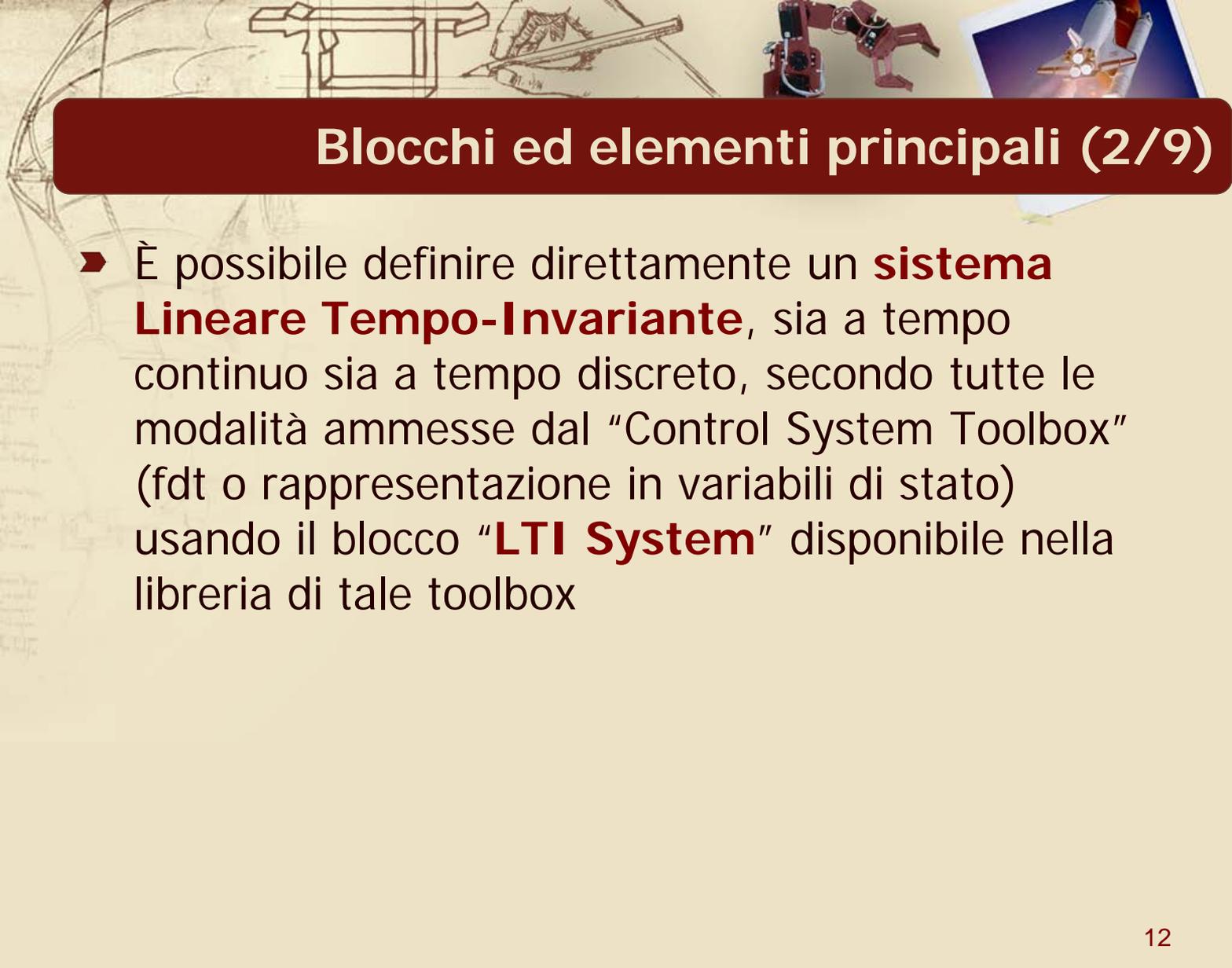
## Creazione di un modello Simulink (4/4)

- Premendo il **tasto destro del mouse** in corrispondenza di un blocco selezionato, si apre un menu a tendina che consente di agire su tutte le caratteristiche del blocco, sia di contenuto (parametri) sia grafiche (font, rotazioni del blocco, colori, ecc.)

# Blocchi ed elementi principali (1/9)

- Nella cartella "**Continuous**" sono disponibili i principali blocchi associati alla rappresentazione di **sistemi dinamici a tempo continuo**, mentre nella cartella "**Discrete**" si possono trovare quelli associati alla rappresentazione di **sistemi dinamici a tempo discreto**

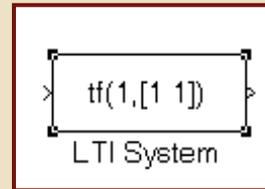
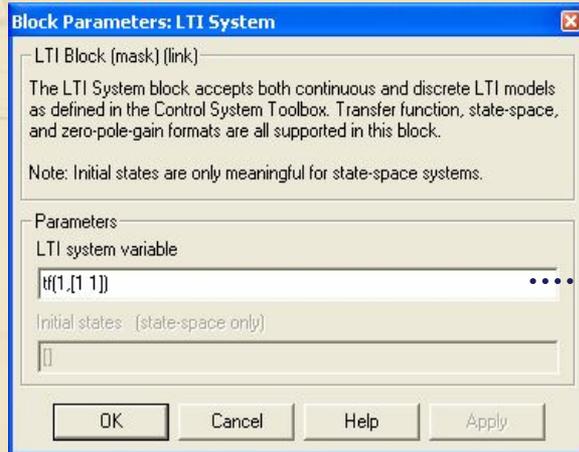




## Blocchi ed elementi principali (2/9)

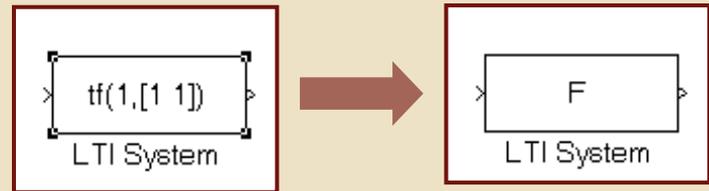
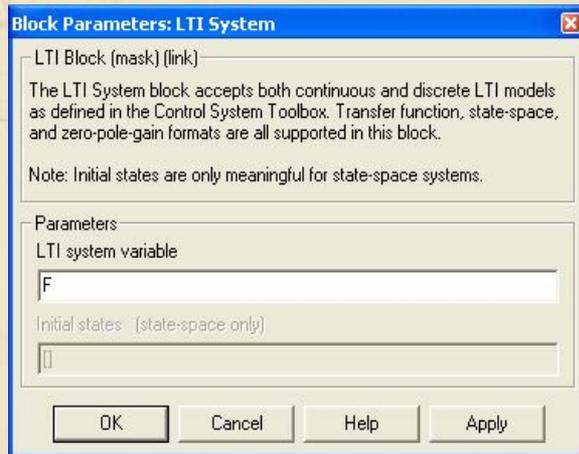
- È possibile definire direttamente un **sistema Lineare Tempo-Invariante**, sia a tempo continuo sia a tempo discreto, secondo tutte le modalità ammesse dal “Control System Toolbox” (fdt o rappresentazione in variabili di stato) usando il blocco “**LTI System**” disponibile nella libreria di tale toolbox

# Blocchi ed elementi principali (3/9)



La definizione della fdt viene automaticamente proposta per mezzo del comando `tf`

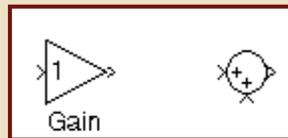
# Blocchi ed elementi principali (3/9)



- Se il sistema è già stato definito nello spazio di lavoro di MATLAB, è sufficiente associare al blocco la sua fdt già calcolata

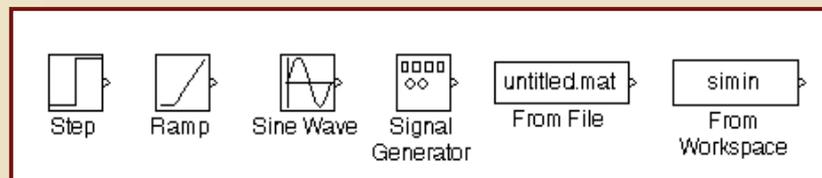
## Blocchi ed elementi principali (4/9)

- Nella cartella "**Math Operations**" si trovano tutti gli elementi che realizzano le operazioni matematiche, tra cui i blocchi "**Gain**" (**guadagno**) e "**Sum**" (**sommatore**)



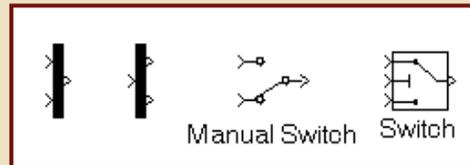
## Blocchi ed elementi principali (5/9)

- Nella cartella "**Sources**" si trovano i blocchi che permettono di applicare un segnale generato a piacere, tra cui "**Step**", "**Ramp**", "**Sine Wave**" e "**Signal generator**", oppure salvato in un file .mat o predefinito nello spazio di lavoro di MATLAB



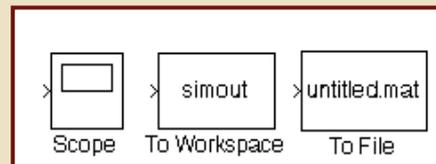
## Blocchi ed elementi principali (6/9)

- Nella cartella "**Signal Routing**" si trovano blocchi utili per la gestione dei segnali, tra cui "**Mux**" e "**Demux**" (per le funzioni di multiplexer e demultiplexer) e gli interruttori manuale ("**Manual Switch**") ed automatico ("**Switch**")



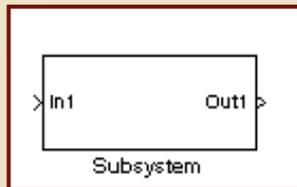
## Blocchi ed elementi principali (7/9)

- Nella cartella "**Sinks**" sono disponibili i blocchi per visualizzare immediatamente una variabile ("**Scope**" ), per renderla disponibile nello spazio di lavoro di Matlab ("**To Workspace**") o per salvarla in un file .mat ("**To File**")

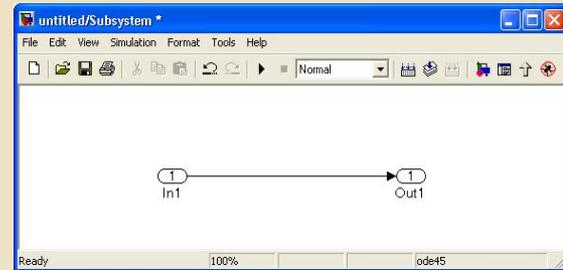
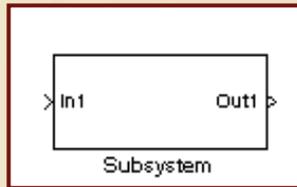


## Blocchi ed elementi principali (8/9)

- È possibile **raggruppare** una parte di un sistema complesso **in un unico blocco di sottosistema** ("Subsystem"), mantenendo inalterati i suoi collegamenti per mezzo di **porte di ingresso e di uscita** ("In" e "Out"), utilizzando gli elementi disponibili nella cartella "**Ports & Subsystems**"



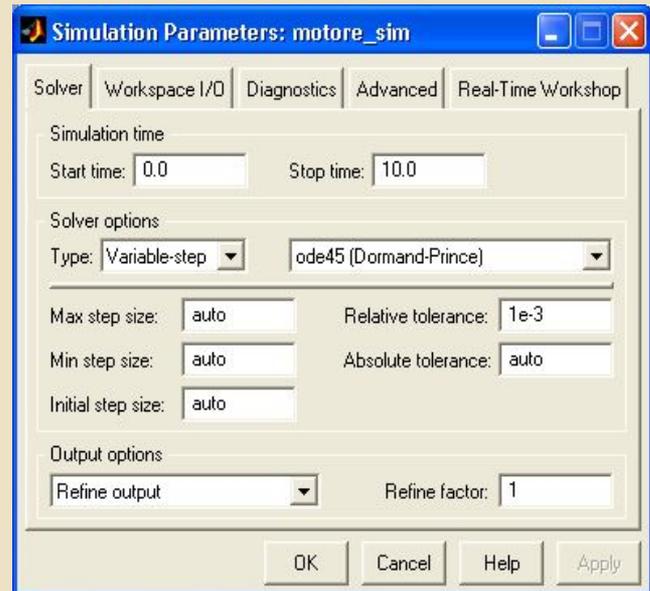
# Blocchi ed elementi principali (9/9)



- Con un doppio click sul blocco di sottosistema, si apre la finestra in cui dovrà essere costruito il suo modello. È possibile inserire ulteriori ingressi e/o uscite aggiungendo ulteriori elementi di **"In"** e **"Out"**

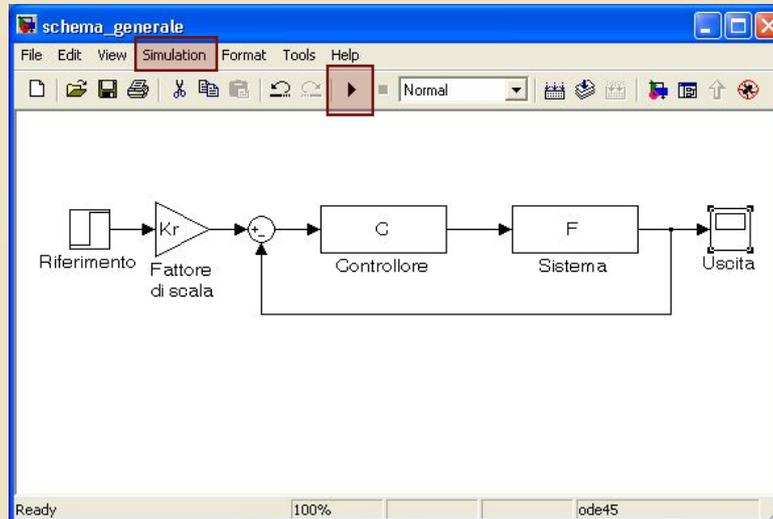
## Esecuzione della simulazione (1/2)

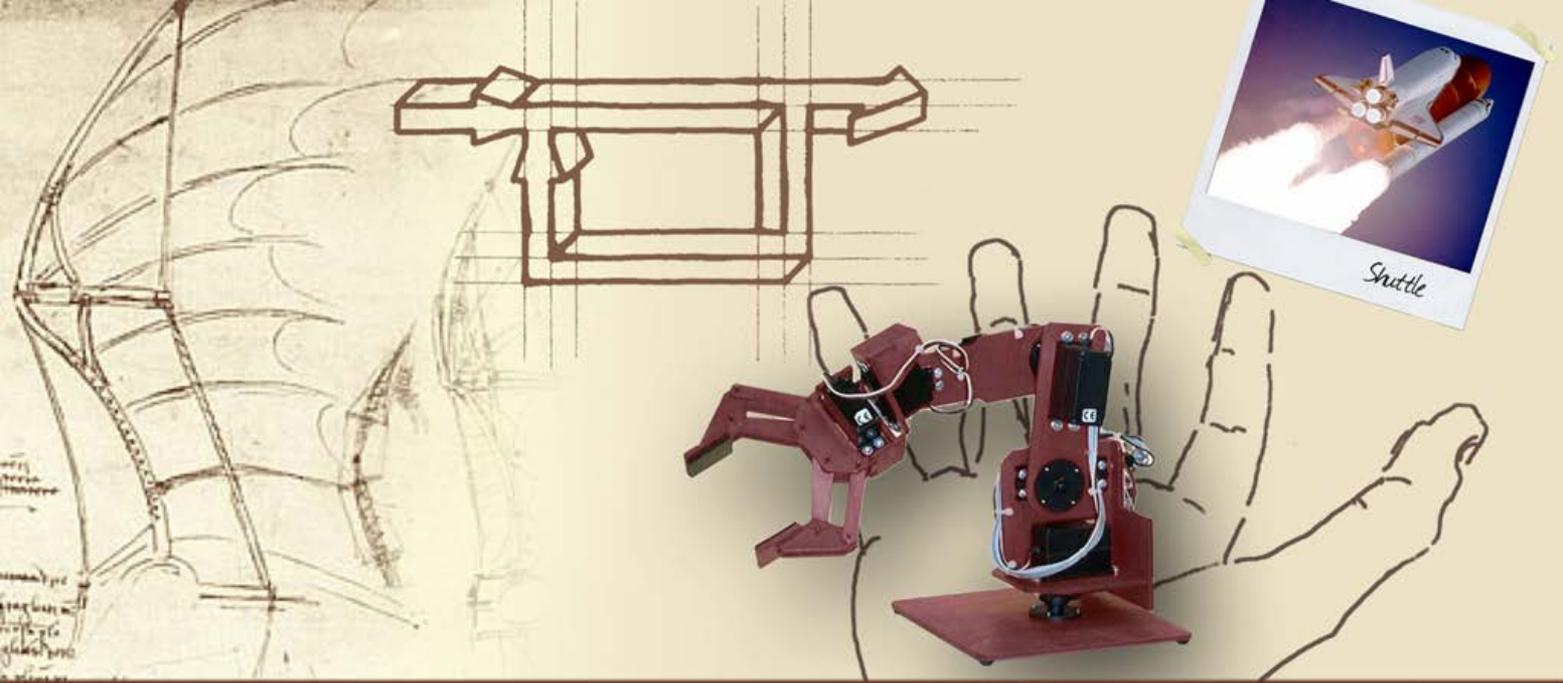
- I **parametri di simulazione** (istante iniziale ed istante finale, algoritmo di integrazione numerica, passo di integrazione e tolleranze sull'errore) possono essere modificati selezionando **"Simulation parameters"** dal menu **"Simulation"**



## Esecuzione della simulazione (2/2)

- Per **eseguire la simulazione**, è sufficiente selezionare "**Start**" dal menu "**Simulation**" (oppure utilizzare il tasto rapido di "Start simulation" sulla barra degli strumenti)



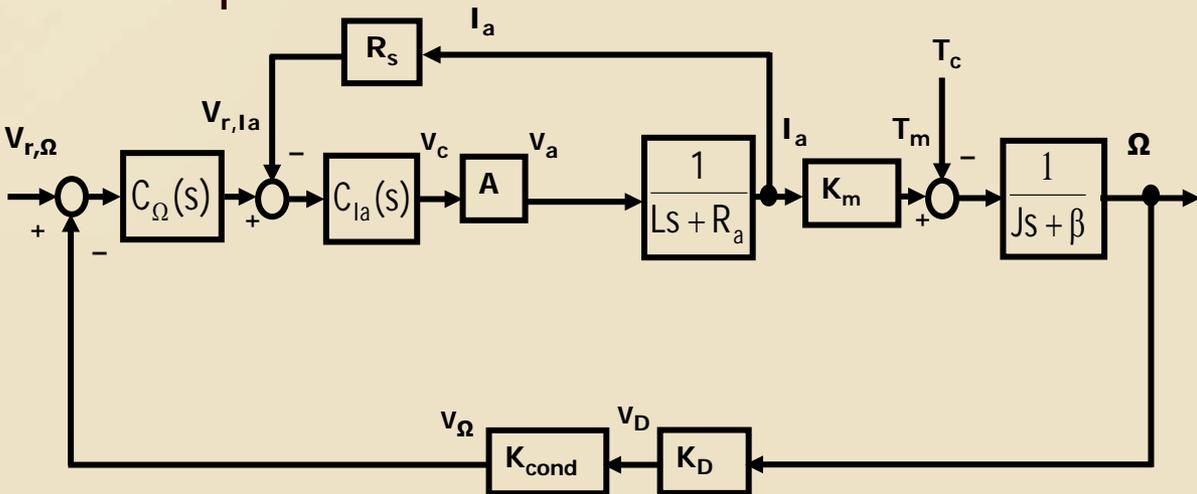


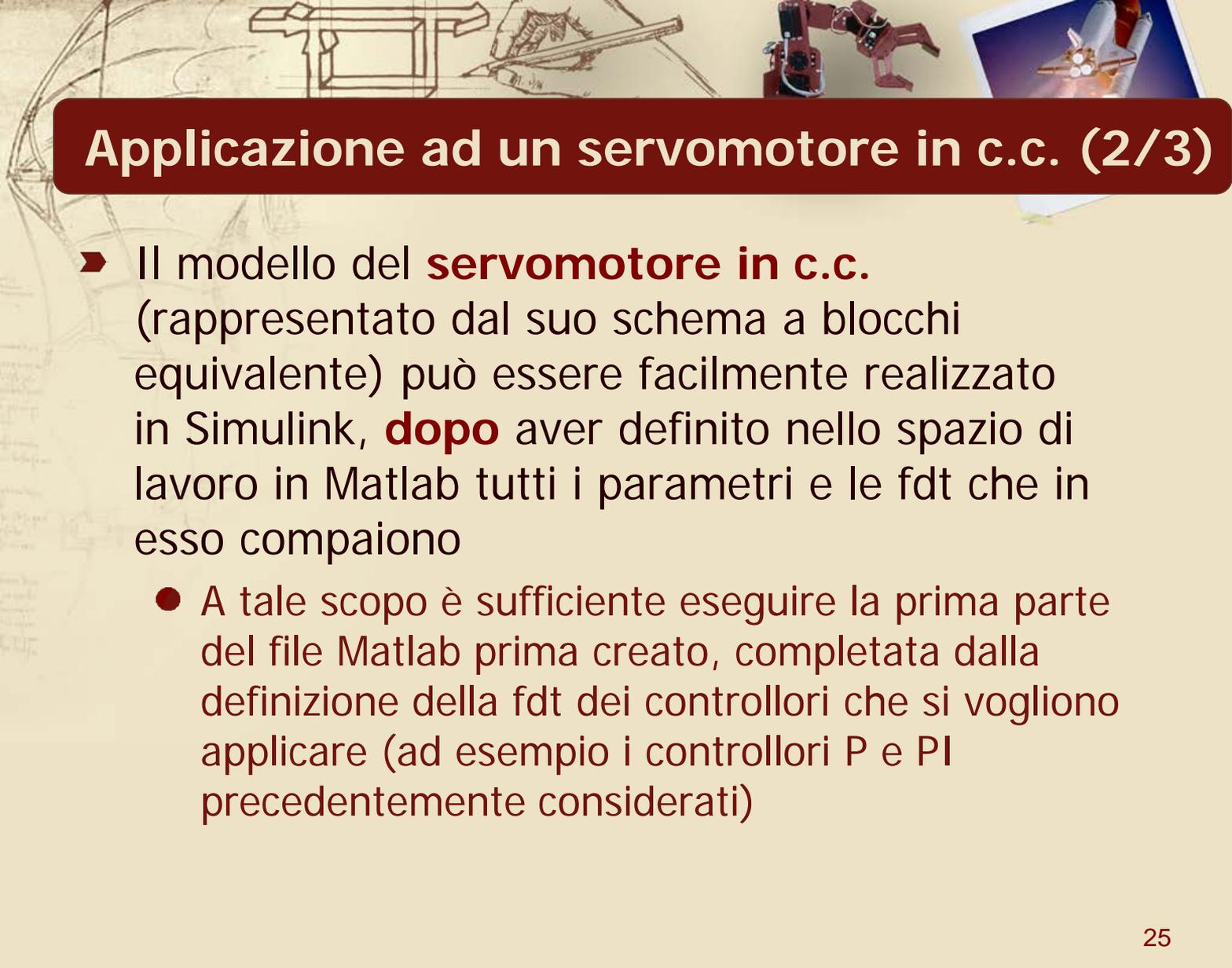
**Matlab e Simulink per i sistemi di controllo**

**Simulazione in ambiente Simulink**

# Applicazione ad un servomotore in c.c. (1/3)

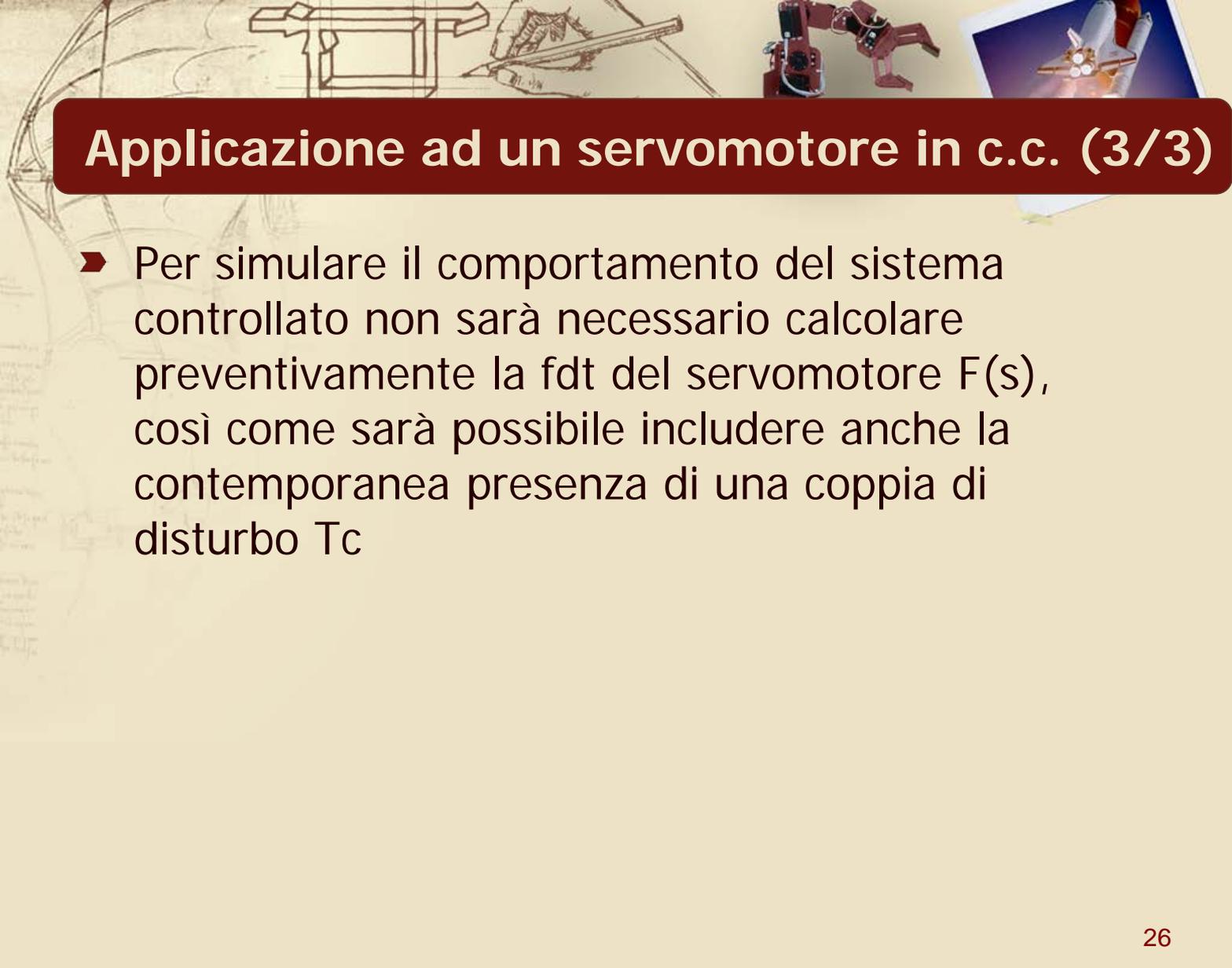
- Il modello del **servomotore in c.c.** (rappresentato dal suo schema a blocchi equivalente) può essere facilmente realizzato in Simulink, **dopo** aver definito nello spazio di lavoro in Matlab tutti i parametri e le fdt che in esso compaiono





## Applicazione ad un servomotore in c.c. (2/3)

- Il modello del **servomotore in c.c.** (rappresentato dal suo schema a blocchi equivalente) può essere facilmente realizzato in Simulink, **dopo** aver definito nello spazio di lavoro in Matlab tutti i parametri e le fdt che in esso compaiono
  - A tale scopo è sufficiente eseguire la prima parte del file Matlab prima creato, completata dalla definizione della fdt dei controllori che si vogliono applicare (ad esempio i controllori P e PI precedentemente considerati)



## Applicazione ad un servomotore in c.c. (3/3)

- Per simulare il comportamento del sistema controllato non sarà necessario calcolare preventivamente la fdt del servomotore  $F(s)$ , così come sarà possibile includere anche la contemporanea presenza di una coppia di disturbo  $T_c$

# File Matlab di definizione del modello

- Dopo l'esecuzione del file, i parametri e le fdt presenti nello spazio di lavoro saranno **automaticamente** riconosciuti dai blocchi Simulink che li contengono ed a ciascuno di essi sarà associata la rispettiva espressione definita in Matlab

```
clear all  
close all  
s=tf('s');  
Ra=6;  
L=3.24e-3;  
Km=0.0535;  
J=20e-6;  
beta=14e-6;  
KD=0.0285;  
Kcond=0.67;  
Rs=7.525;  
A=2.925;  
K=1000;  
Cla=K/s;
```

Prima parte del file: definizione dei parametri del servomotore

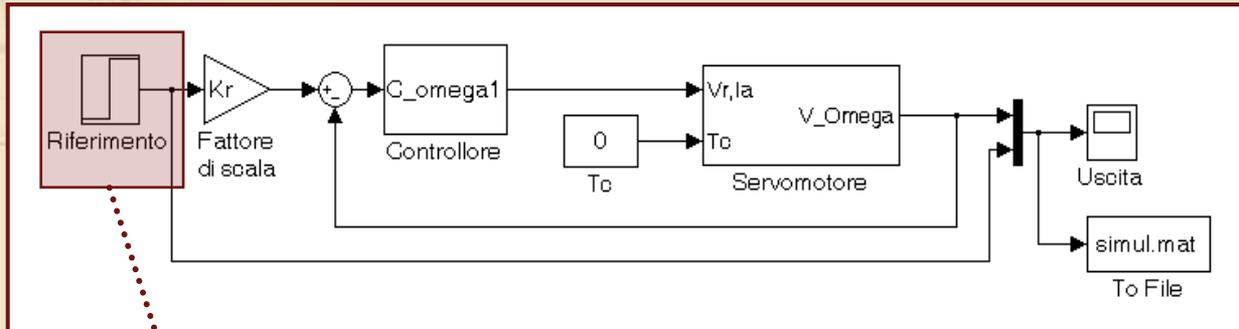
## File Matlab di definizione del modello

- Dopo l'esecuzione del file, i parametri e le fdt presenti nello spazio di lavoro saranno **automaticamente** riconosciuti dai blocchi Simulink che li contengono ed a ciascuno di essi sarà associata la rispettiva espressione definita in Matlab

```
Kr=1;  
Kp=0.4;  
C_omega1=Kp;  
Ki=2;  
C_omega2=Kp+Ki/s;
```

Seconda parte del file:  
definizione dei controllori P e PI

# Modello Simulink del servomotore (1/7)



Blocco Step  
da Sources



Block Parameters: Riferimento

Step  
Output a step.

Parameters:

Step time:  
0

Initial value:  
0

Final value:  
1

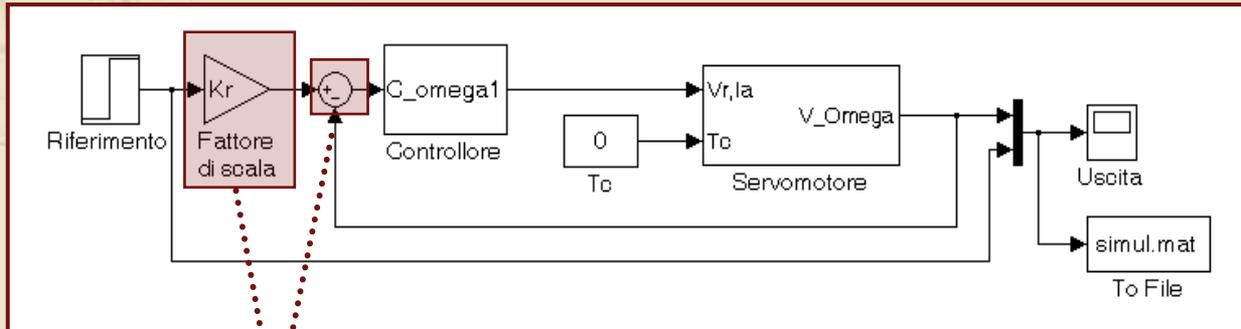
Sample time:  
0

Interpret vector parameters as 1-D

Enable zero crossing detection

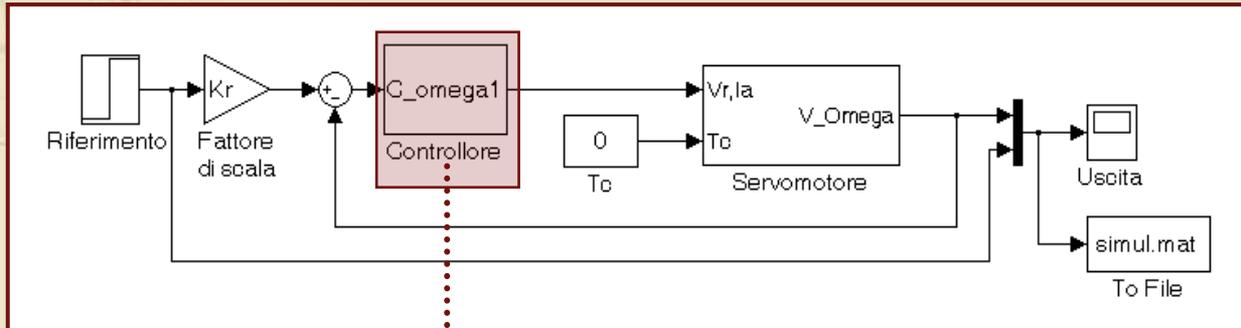
OK Cancel Help Apply

# Modello Simulink del servomotore (2/7)



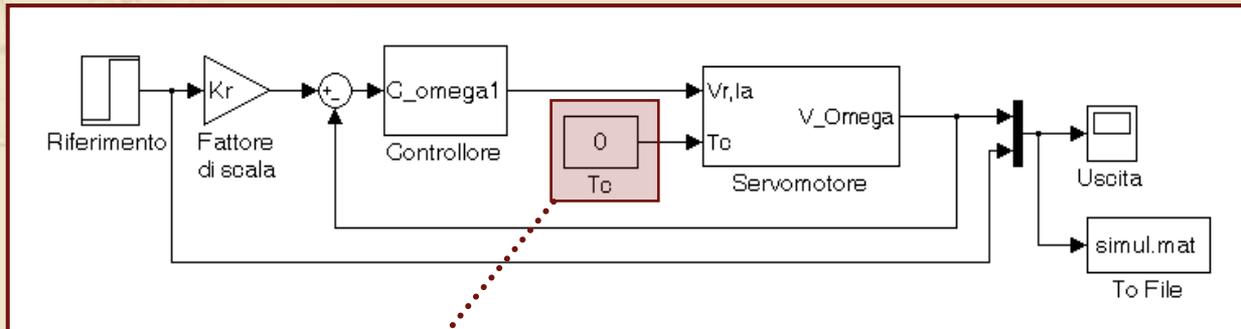
Blocchi **Gain** e **Sum**  
da **Math**  
**Operations**

# Modello Simulink del servomotore (3/7)



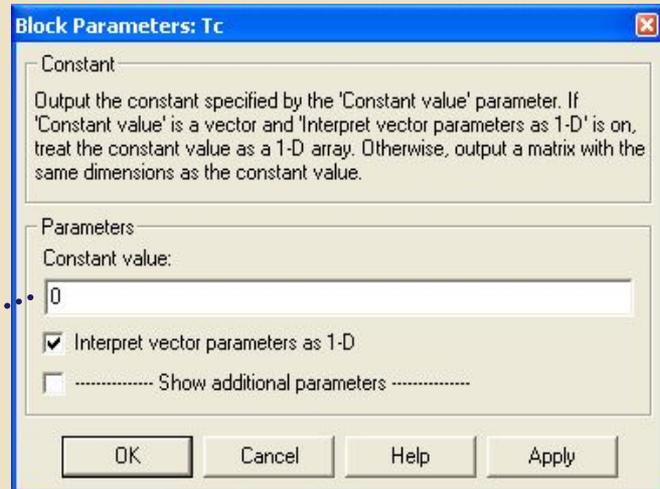
Blocco LTI System da  
Control System  
Toolbox

# Modello Simulink del servomotore (4/7)

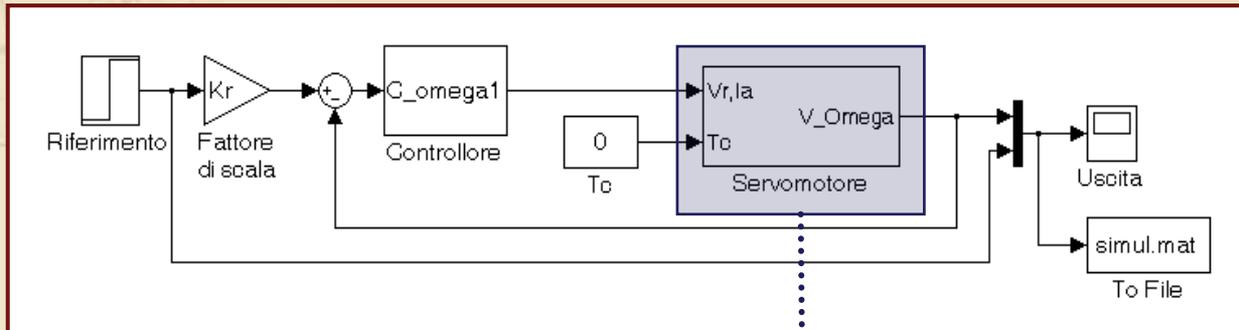


Blocco  
**Constant** da  
Sources

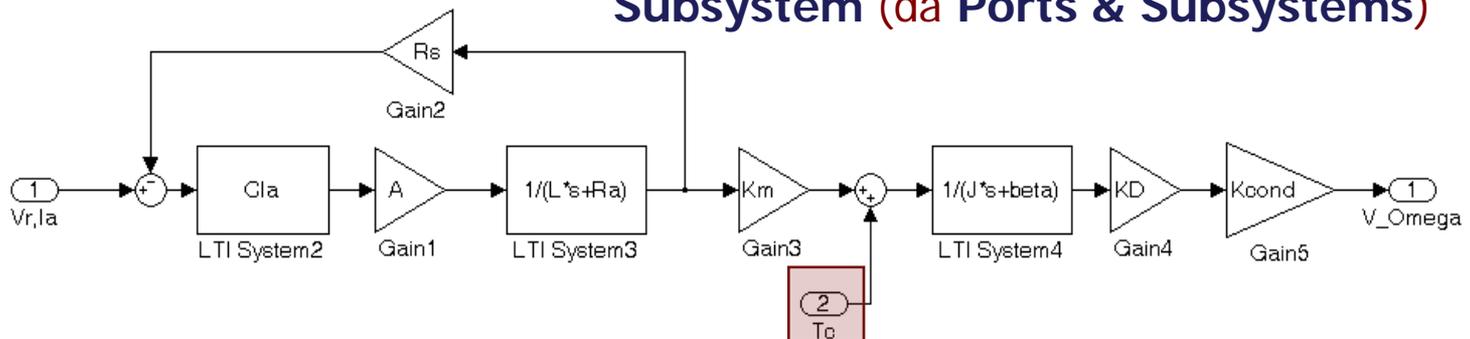
Il disturbo è imposto  
nullo come nella  
simulazione in Matlab



# Modello Simulink del servomotore (5/7)

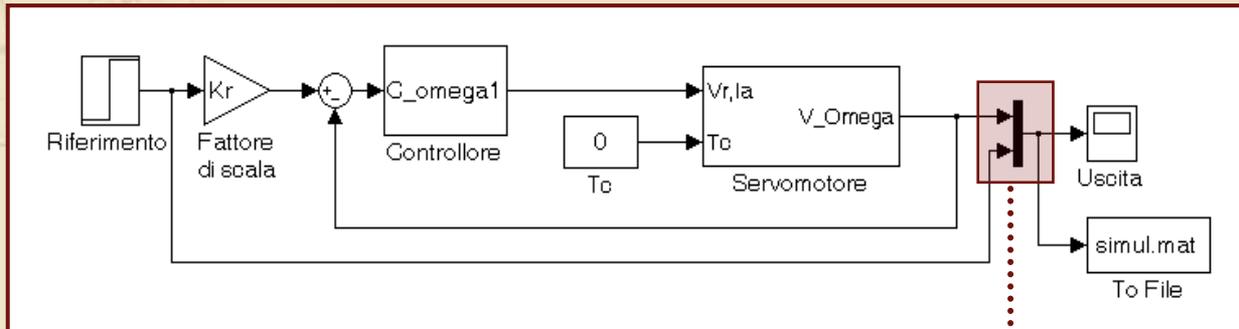


## Subsystem (da Ports & Subsystems)



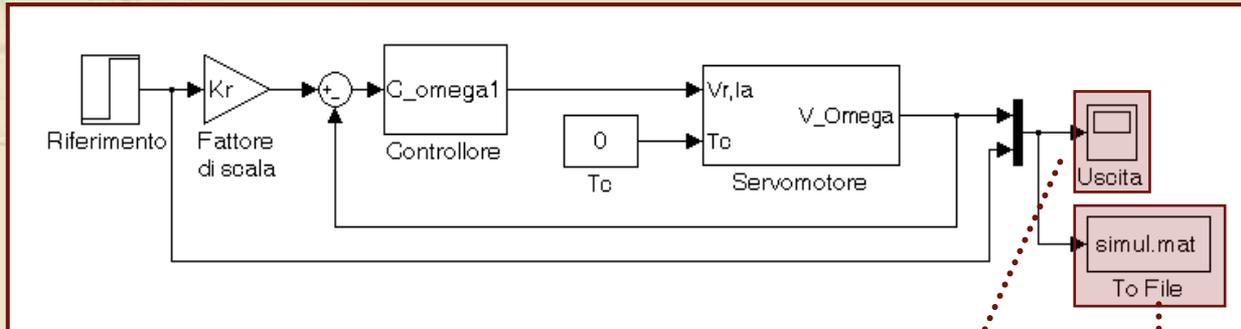
Blocco **In** aggiuntivo rispetto agli **In** e **Out** preesistenti

# Modello Simulink del servomotore (6/7)



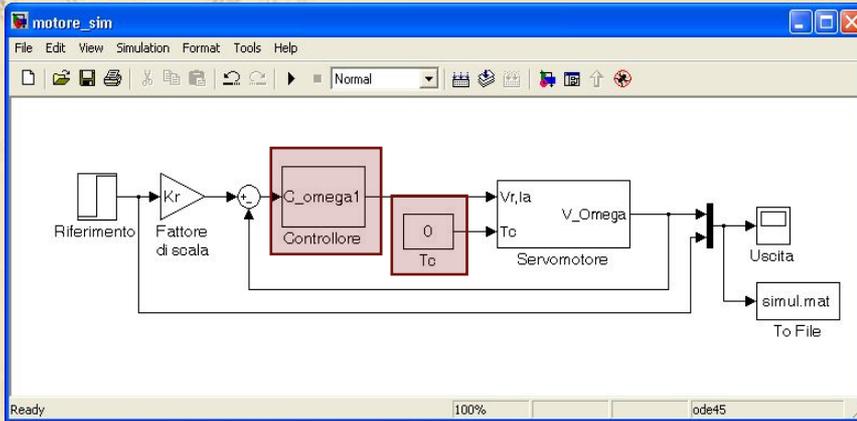
**Blocco Mux da  
Signal  
Routing**

# Modello Simulink del servomotore (7/7)



Blocchi **Scope** e  
**To File** da  
**Sinks**

# Simulazione 1: controllore P, $T_c = 0$ (1/3)



Parametri di simulazione

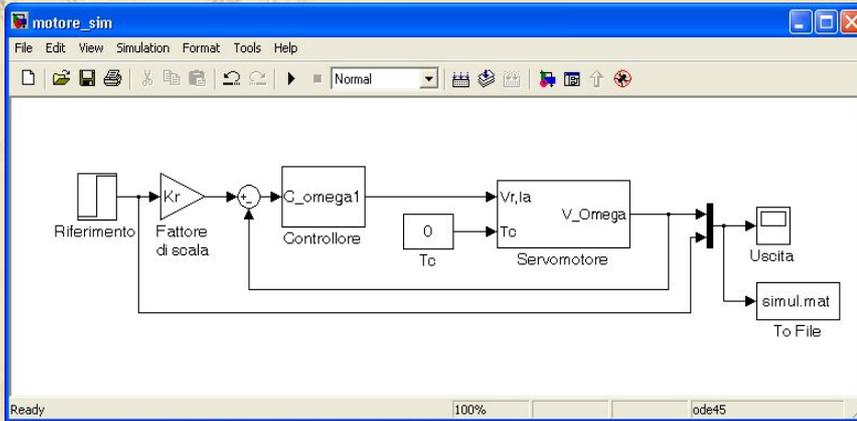


The dialog box shows the following settings:

- Solver: Workspace I/O | Diagnostics | Advanced | Real-Time Workshop
- Simulation time: Start time: 0.0, Stop time: 10.0
- Solver options: Type: Variable-step, ode45 (Dormand-Prince)
- Max step size: auto, Relative tolerance: 1e-3
- Min step size: auto, Absolute tolerance: auto
- Initial step size: auto
- Output options: Refine output, Refine factor: 1

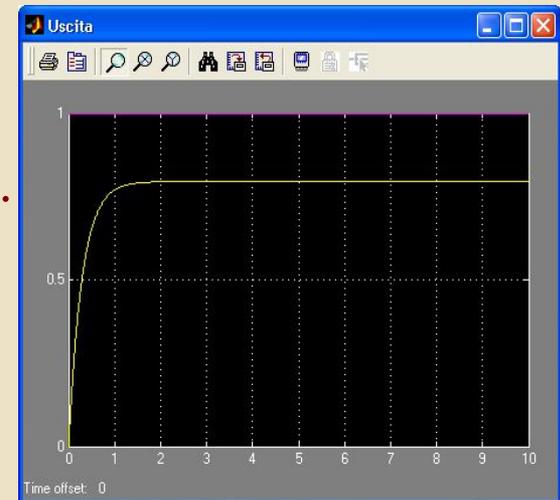
Buttons: OK, Cancel, Help, Apply

# Simulazione 1: controllore P, $T_c = 0$ (2/3)

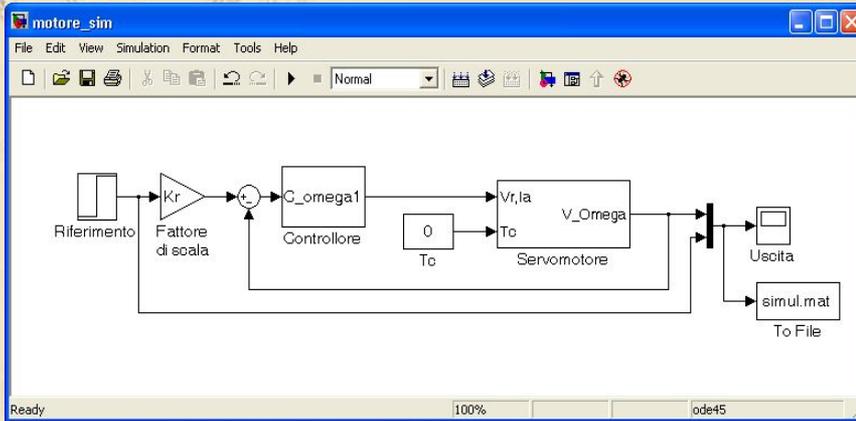


Risultato visibile  
sull'oscilloscopio

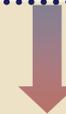
Conferma del risultato  
ottenuto con Matlab:  
**elevato errore  
finale**



# Simulazione 1: controllore P, $T_c = 0$ (3/3)



Creazione del file  
"simul.mat" contenente  
la variabile "simulazione"



**Block Parameters: To File**

To File  
Write time and input to specified MAT file in row format. Time is in row 1.

Parameters

Filename:  
simul.mat

Variable name:  
simulazione

Decimation:  
1

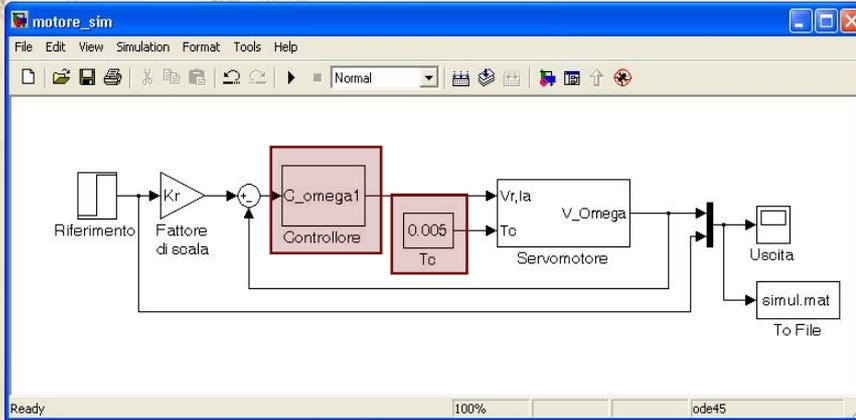
Sample time (-1 for inherited):  
-1

OK Cancel Help Apply

La variabile "simulazione" generata ha tre righe:

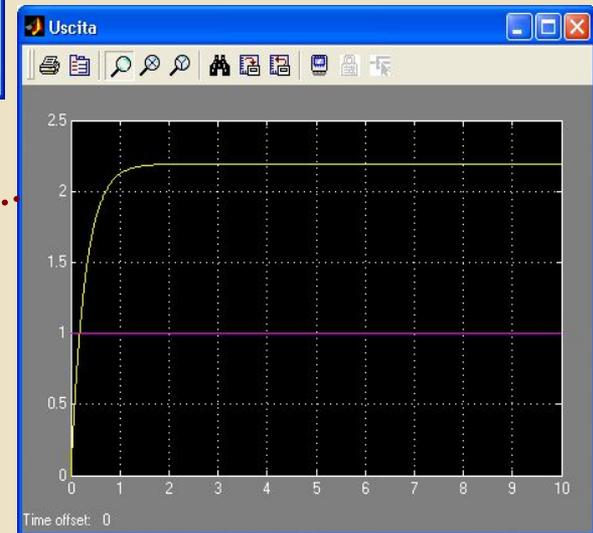
- 1) campioni del tempo
- 2) campioni dell'uscita
- 3) campioni del riferimento

# Simulazione 2: controllore P, $T_c = 0.005$

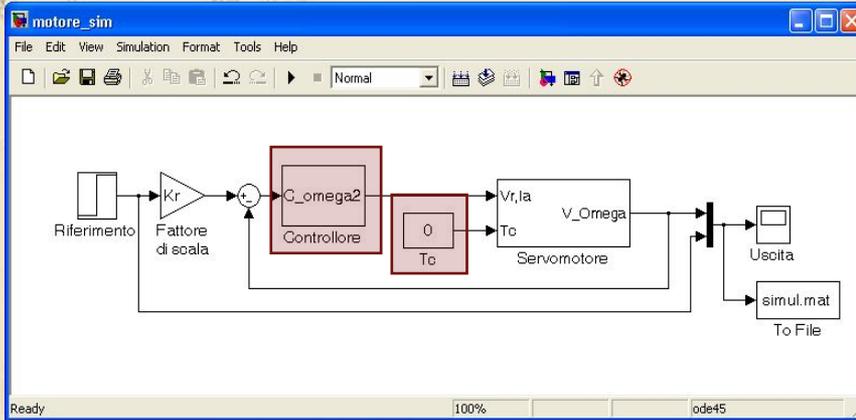


Risultato visibile  
sull'oscilloscopio

Si riscontra una  
significativa **variazione**  
dell'errore finale

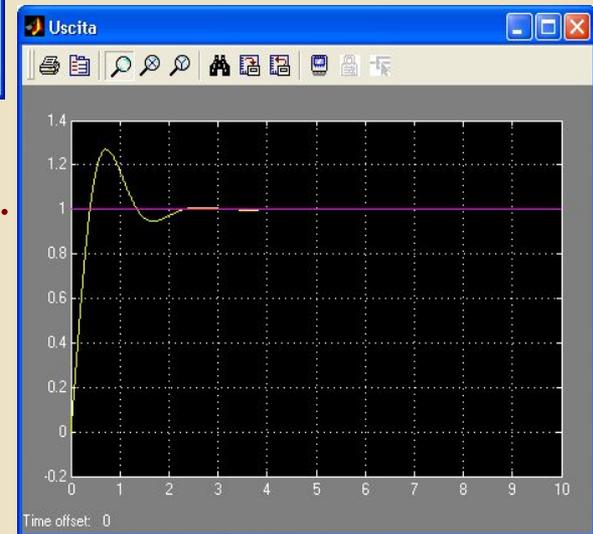


# Simulazione 3: controllore PI, $T_c = 0$

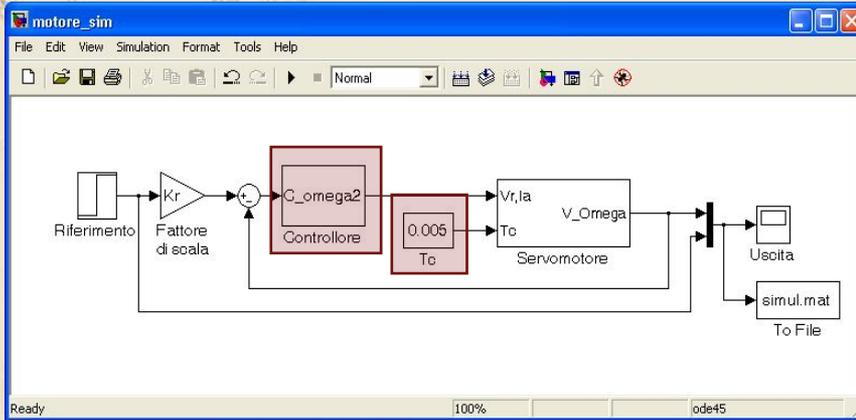


Risultato visibile  
sull'oscilloscopio

Conferma del risultato  
ottenuto con Matlab:  
presenza di  
**significativa  
sovraelongazione,  
errore finale nullo**



# Simulazione 4: controllore PI, $T_c = 0.005$



Risultato visibile  
sull'oscilloscopio



Si riscontra un elevato  
aumento della  
sovraelongazione,  
mentre l'errore finale  
rimane nullo

