Systems and Control Theory

Master Degree Course in ELECTRONICS ENGINEERING

http://www.dii.unimore.it/~lbiagiotti/SystemsControlTheory.html

Introduction to SIMULINK

Luigi Biagiotti

e-mail: luigi.biagiotti@unimore.it

http://www.dii.unimore.it/~lbiagiotti

Simulink introduction

- **Simulink** (**Simu**lation and **Link**) is an extension of MATLAB that offers modeling, simulation, and analysis of dynamical systems under a **graphical user interface** (GUI) environment.

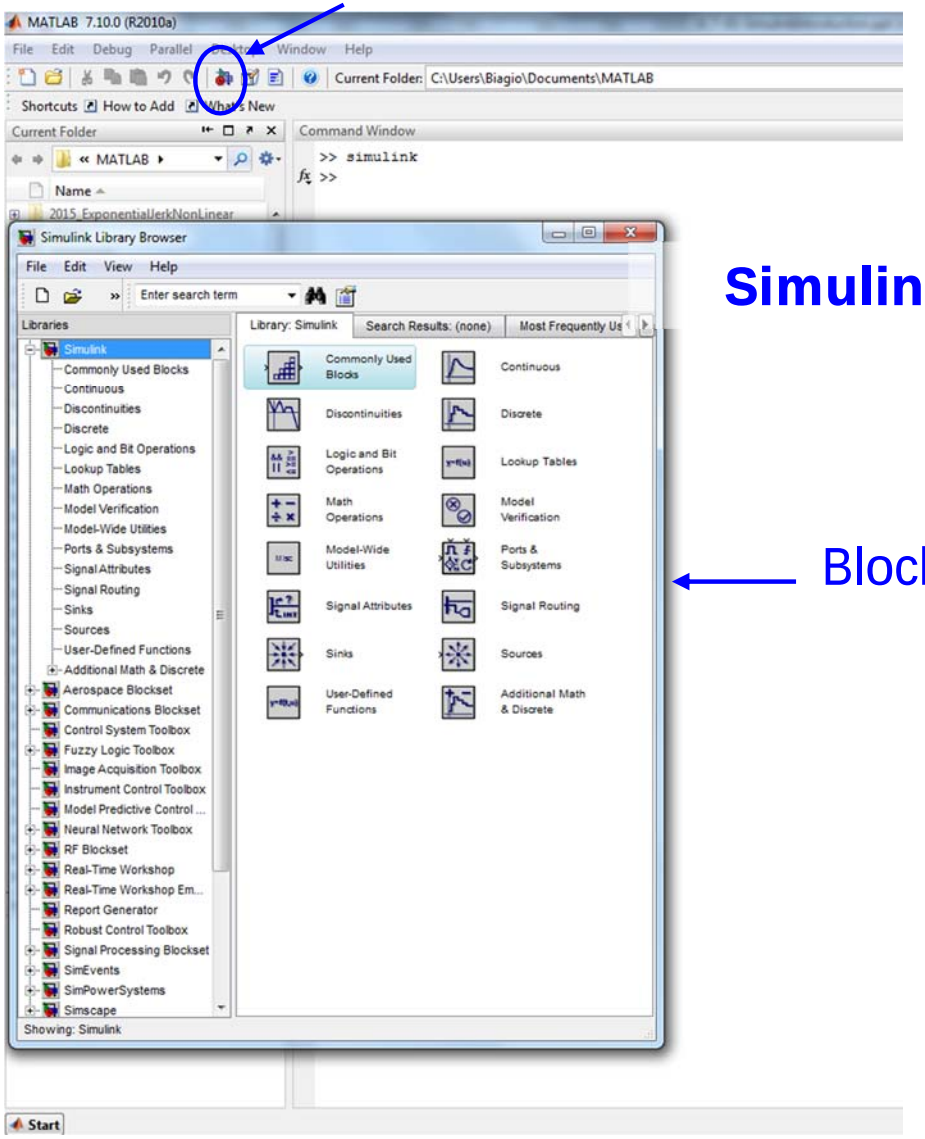
- Simulink is based on block diagrams of Dynamic Systems

- The construction of a model is simplified with click-and-drag mouse operations. Simulink includes a comprehensive block library of toolboxes for both linear and nonlinear analyses.

- As Simulink is an integral part of MATLAB, it is easy to switch back and forth during the analysis process and thus, the user may take full advantage of features offered in both environments.

Starting Simulink

- From Matlab command window enter **simulink**, or alternately, click on the **Simulink icon** located on the toolbar



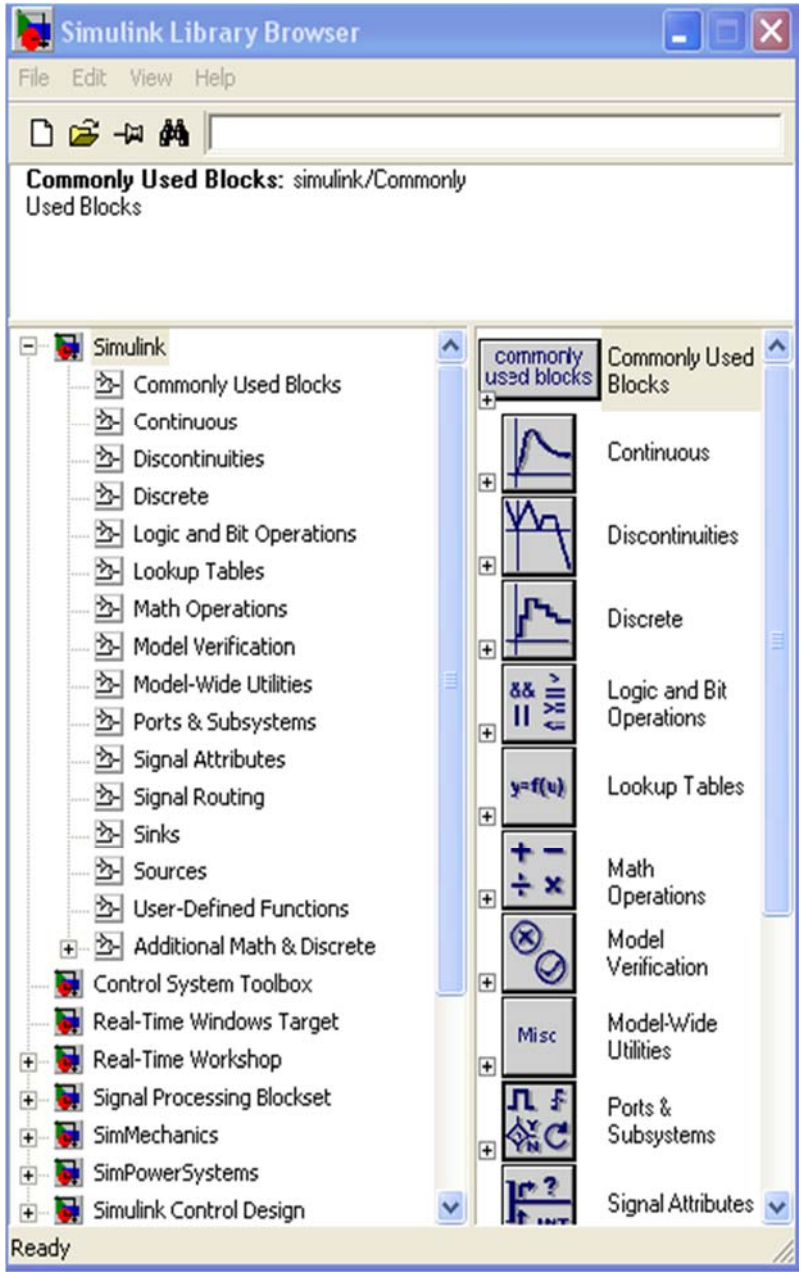
Simulink's library browser

Library →

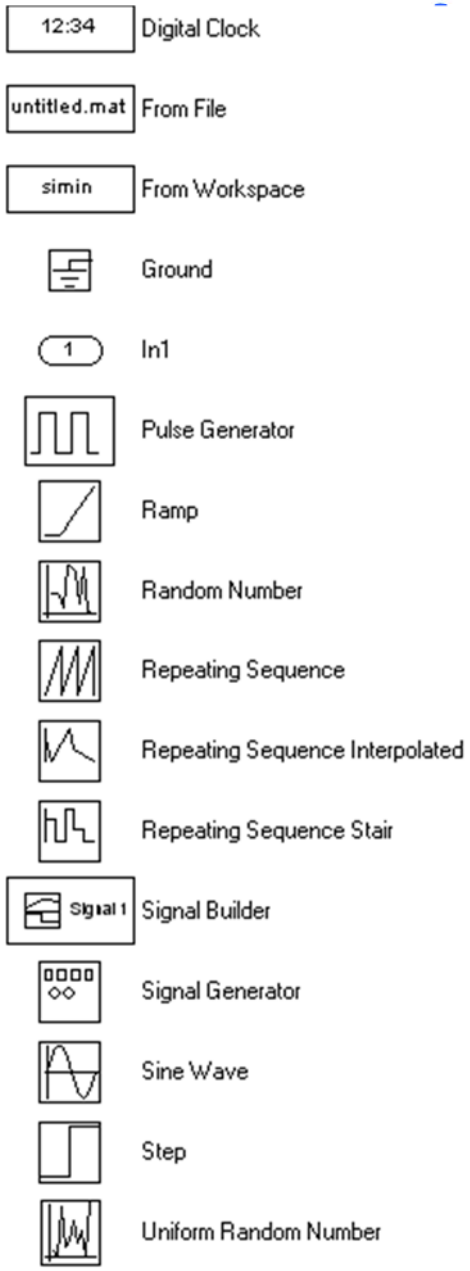
← Block set

Simulink library

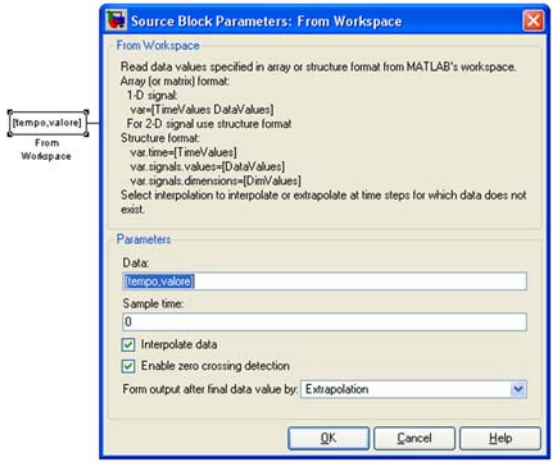
- The block library is organized into **functional groups**. For instance
  - *Sources:* Blocks for the generation of input signals (steps, sinusoids etc.)
  - *Sinks:* Blocks for the graphical visualization of signals
  - *Math:* Blocks for the mathematical elaboration of signals
  - *Continuous:* Blocks for the definition of (continuous) transfer functions



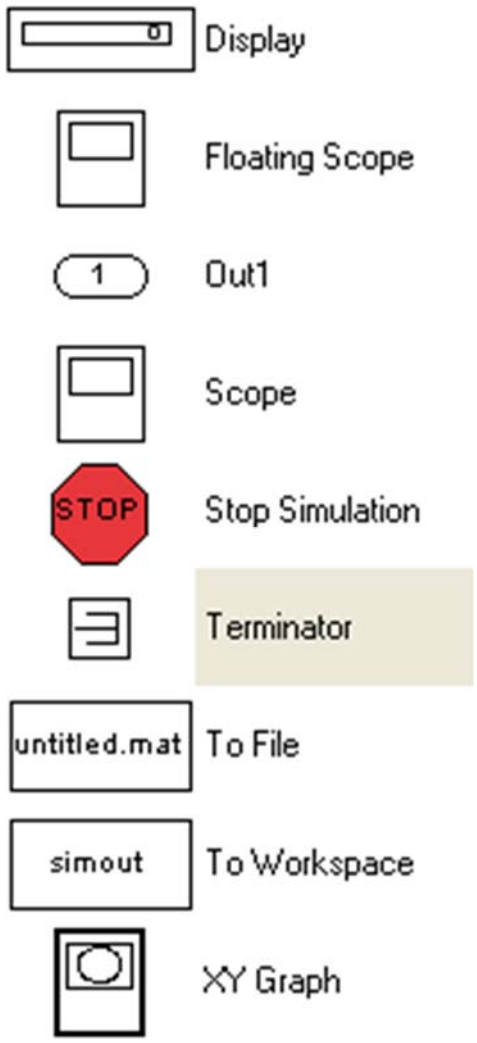
Simulink – Sources Library



- The most used blocks are:
  - **Constant:** for generating a constant value
  - **Step:** for generating a step function
  - **Ramp:** for generating a ramp function
  - **Sine wave:** for generating a sinusoidal function
  - **From workspace:** the reference signal, previously generated in the MATLAB workspace, is passed as **`[time, value]`**, where **`time`** and **`value`** are two column vectors with the same length
  - **Clock:** it outputs the current simulation time at each simulation step



Simulink – Sinks Library



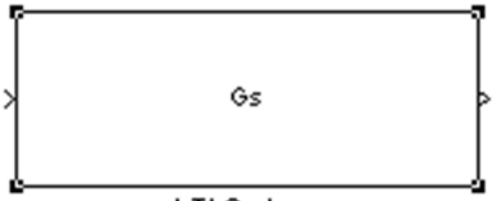
- Set of tools for displaying the output of the simulations
- The most used blocks are:
  - **Scope:** it displays the input signal as a function of time (take care to the option **`limit data points to`**…)
  - **XYGraph:** it produce a graphics of the signal y (on the second input) as a function of the signal x (on the first input).
  - **To Workspace:** it saves the samples of the input signal in a MATLAB variable (N.B.: **`save format array`**).

Simulink – Transfer functions

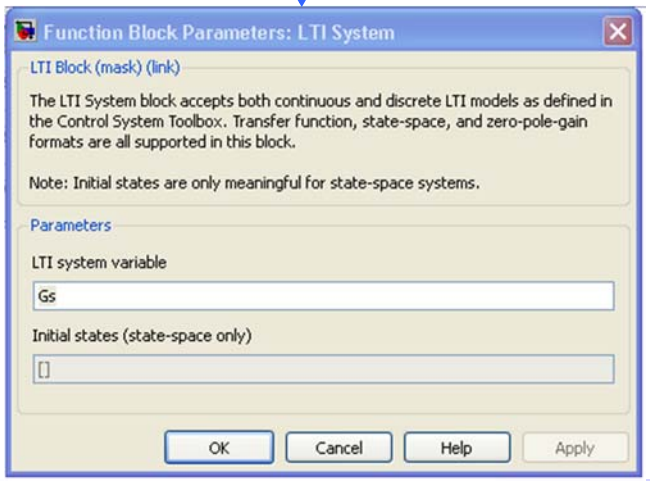
- In order to insert a transfer function in a Simulink scheme the blocks of the library **Continuous** can be used:
  - **Transfer Fcn:** it allows to define a transfer function by specifying the vectors containing the coefficient of numerator and denominator.
  - **Zero-Pole:** it defines a transfer function by specifying the vectors containing polse and zeros of the transfer function.
- The library **Discrete** contains the corresponding blocks for the definition of discrete-time transfer functions.

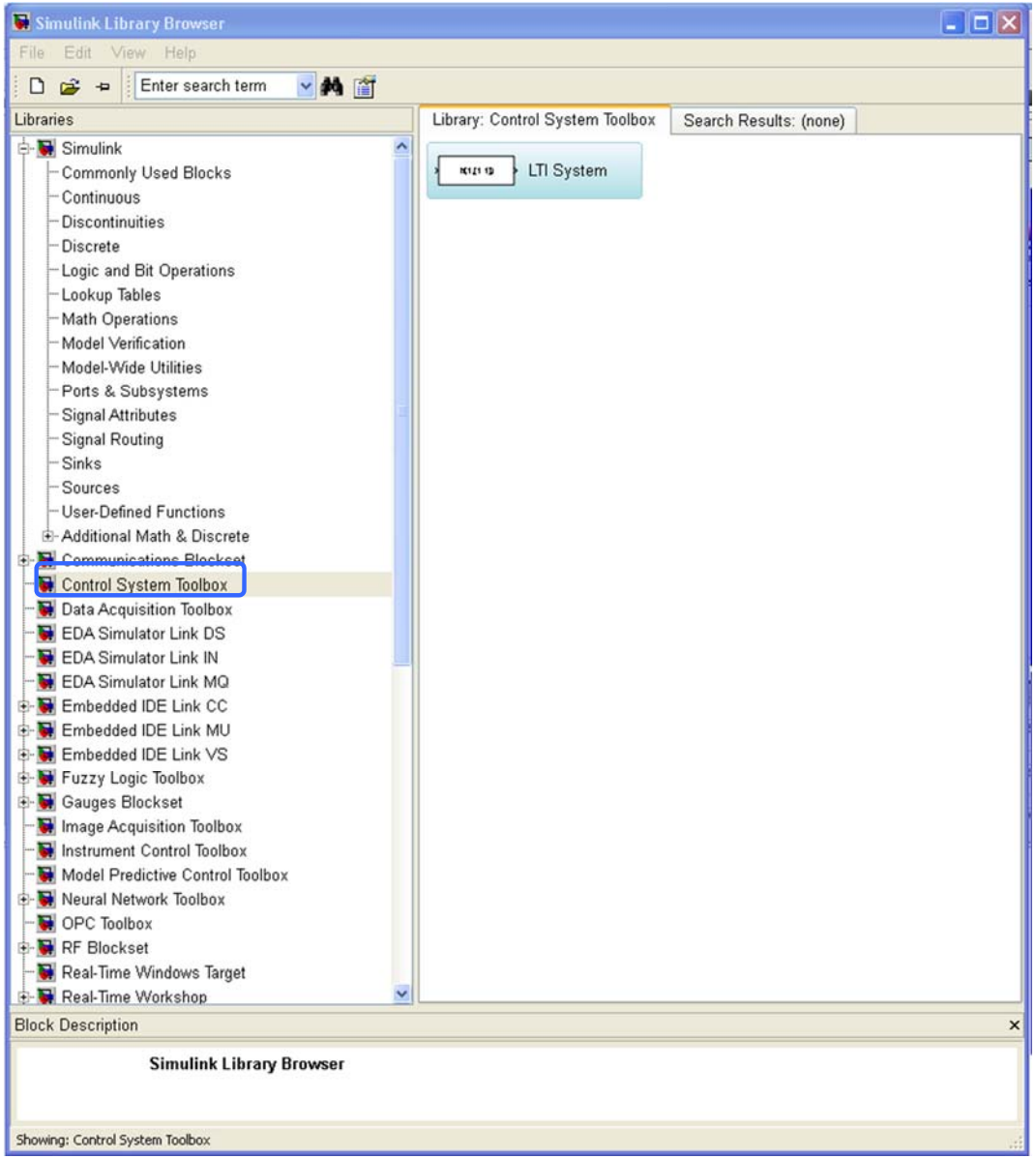
Simulink and the Control System Toolbox

- The Control system Toolbox provides a Simulink block for directly using a transfer function defined in the MATLAB workspace (see command **`tf`**) or a state space model



Double click

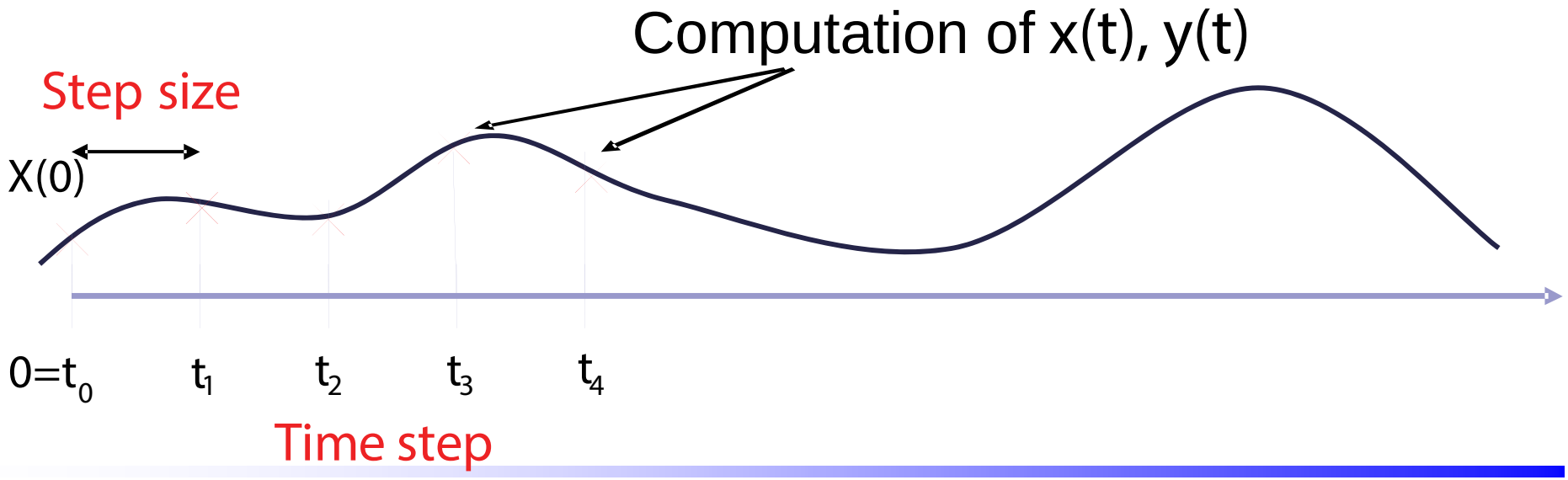




Simulink – Simulation parameters

- Simulink solves ordinary differential equations (e.g. defined by means of transfer functions) by using **techniques for numerical integration**

- Simulating a dynamic system means computing the evolution of the state and output variables for a given time period
- The state and the output values are computed at given time-instants (called *time steps*) which are separeted by integration intervals (called *step sizes*)



Simulink – Methods for numerical integration

Several methods (the so-called *solvers)* for the numerical integration of differential equations exist:

- Integration methods with **fixed step** size (useful for the simulation of discrete-time systems)
  - The smaller the step size is, the more accurate the simulation results. On the other hand, the duration of the simulation grows.
- Integration methods with **variable step** size
  - the solver determines the optimal step size during the simulation (for instance if the system is characterized by fast dynamics, the step size must be reduced accordingly)

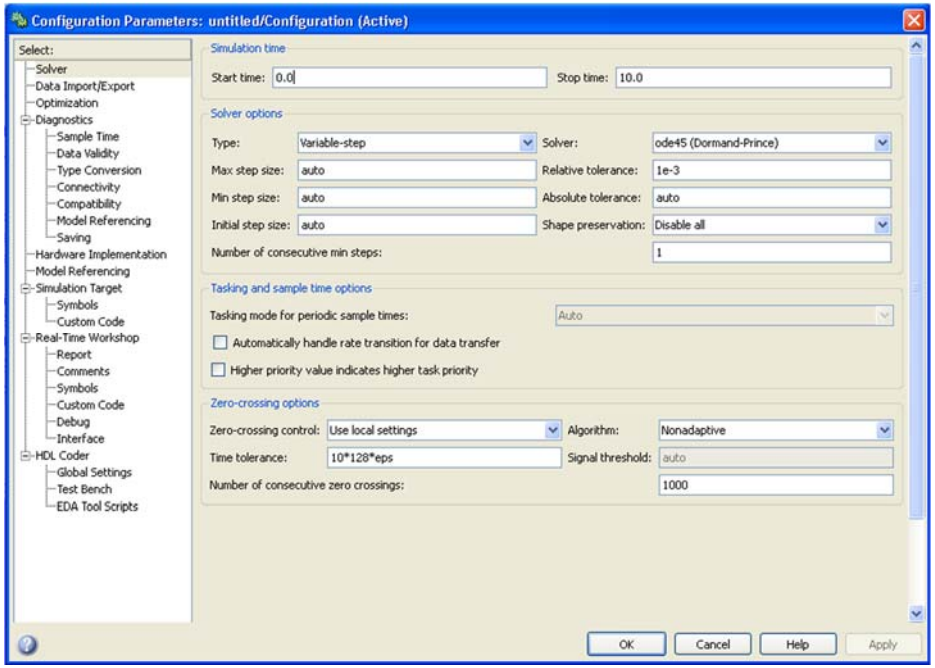
- **ODE45 is the default solver**

Simulation parameters

- Initial and final simulation time-instants
- Solver type (e.g. variable step, ODE45)
- **Max step size** (maximum size of the interval beetwen two successive time-steps)

  Must be chosen:

  - Smaller than the fastest time-constant of the system
  - Smaller than the period of the fastest periodic signal acting on the system (e.g. 1/10 smaller)



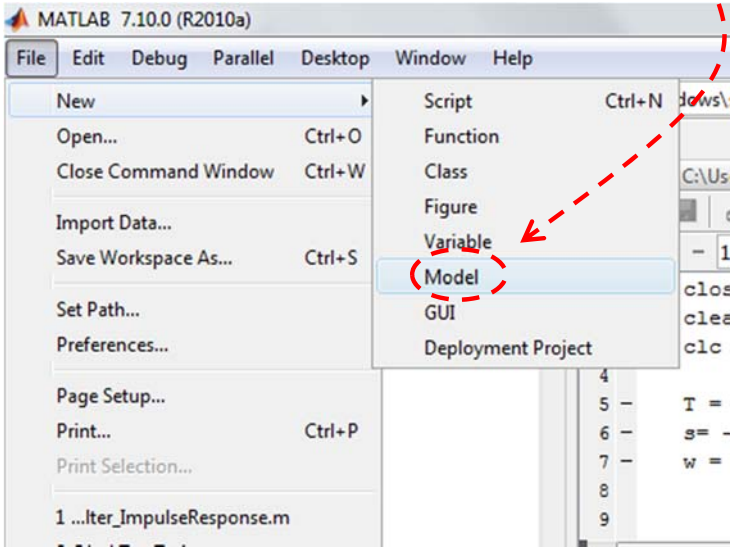
- **Min step size** (minimum size of the interval beetwen two successive time-steps)

  Can be used to reduce the simulation time (at the expense of the accuracy)

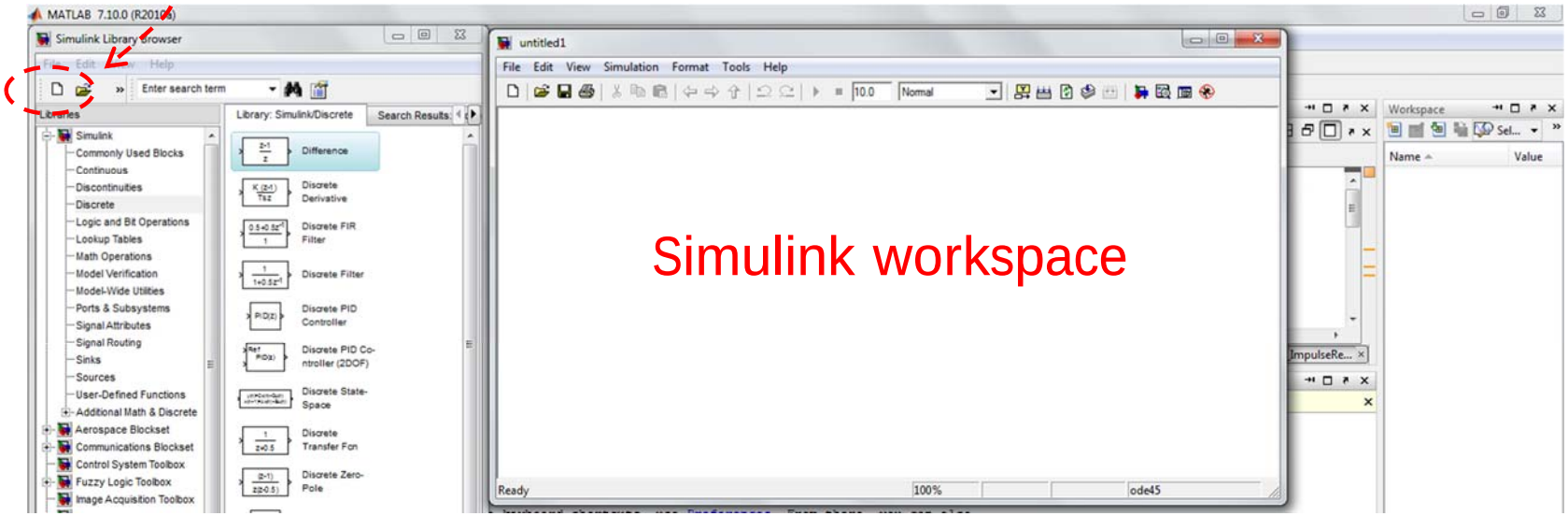
- **Relative/absolute tollerance** Accuracy of the simulation output (set “auto” or values smaller enough, e.g. 1e-4)

Steps for constructing a Simulink model

- Open a new model from the Matlab menu **file**

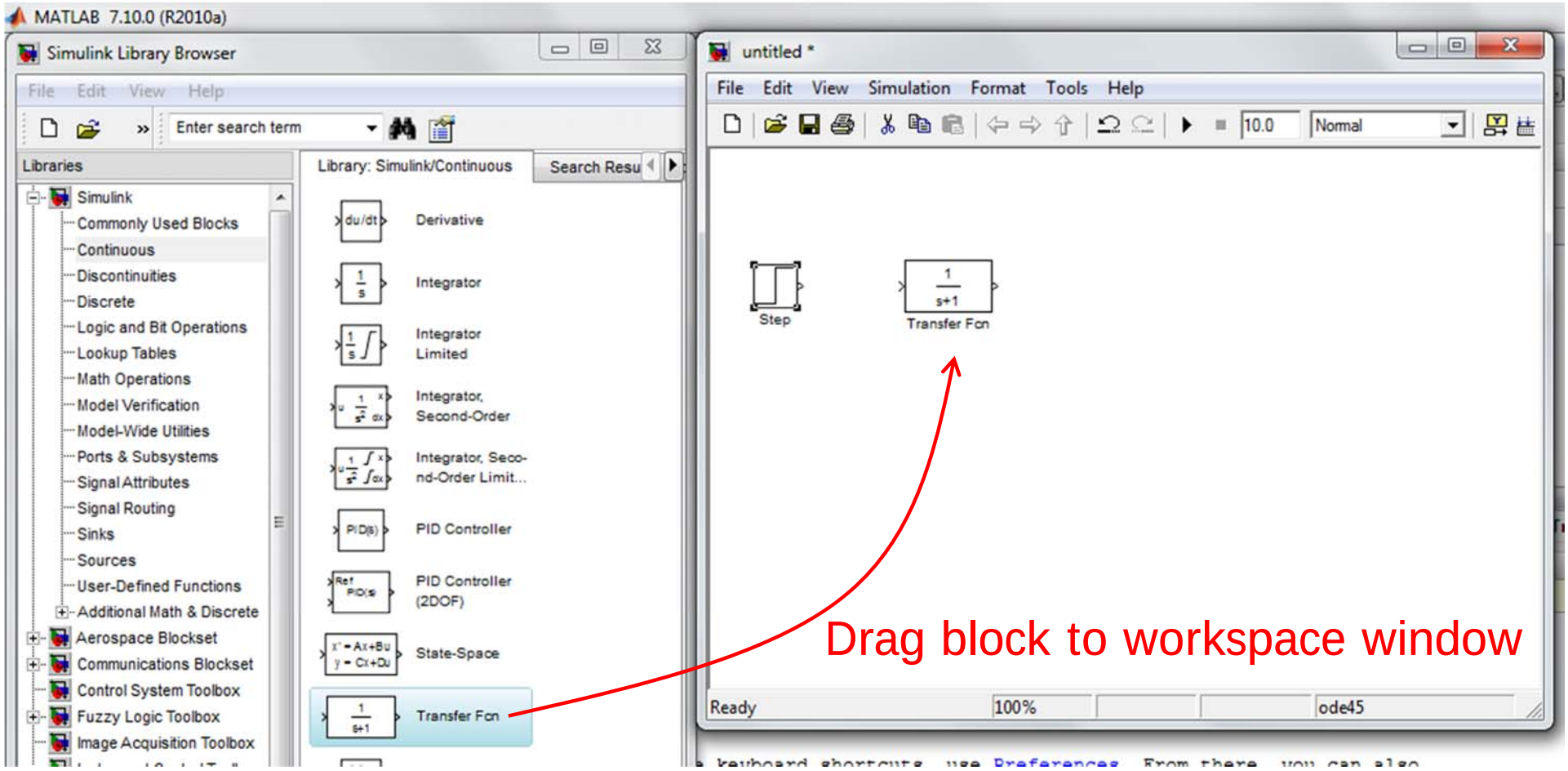


or from the icon in the **Simulink Library browser**



Steps for constructing a Simulink model

- Import blocks in the Simulink workspace



- Connect blocks

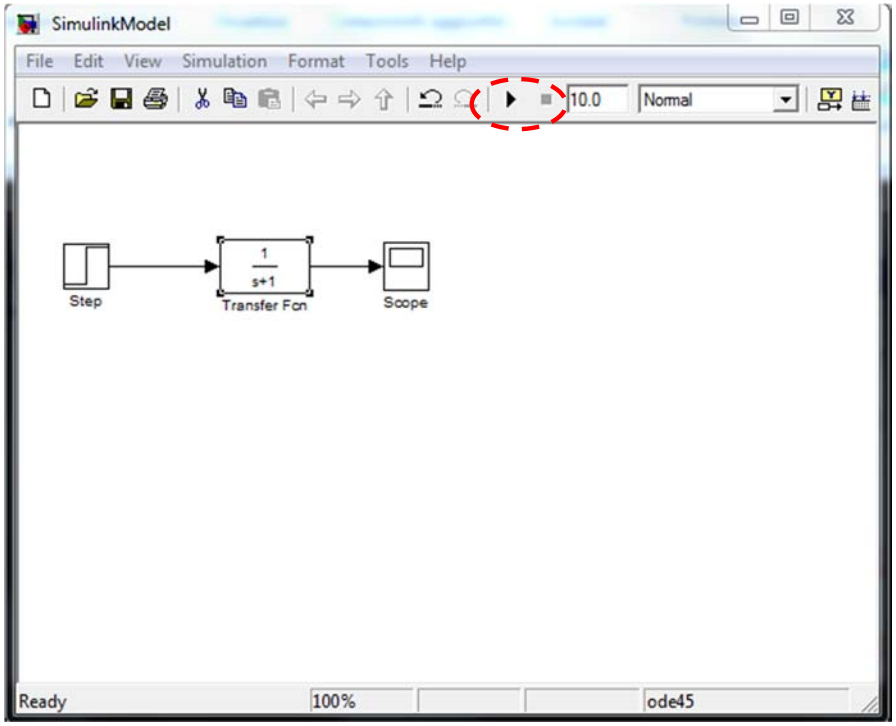
Steps for constructing a Simulink model

- Action on blocks

Actions	Keystrokes or Mouse Actions
Copying a block from a library	Drag the block to the model window with the left button on the mouse OR use select the COPY and PASTE from EDIT menu.
Duplicating blocks in a model	Hold down the CTRL key, select the block with the left mouse button and drag the block to a new location.
Display block's parameters	Double click on the block
Flip a block	CTRL-I
Rotate a block (clockwise 90 deg @ each keystroke)	CTRL-R
Changing blocks' names	Click on block's label and position the cursor to desired place.

Steps for constructing a Simulink model

- Save and run the simulation from the workspace window



or directly from MATLAB

```
>> sim('SimulinkModel',EndTime)
```

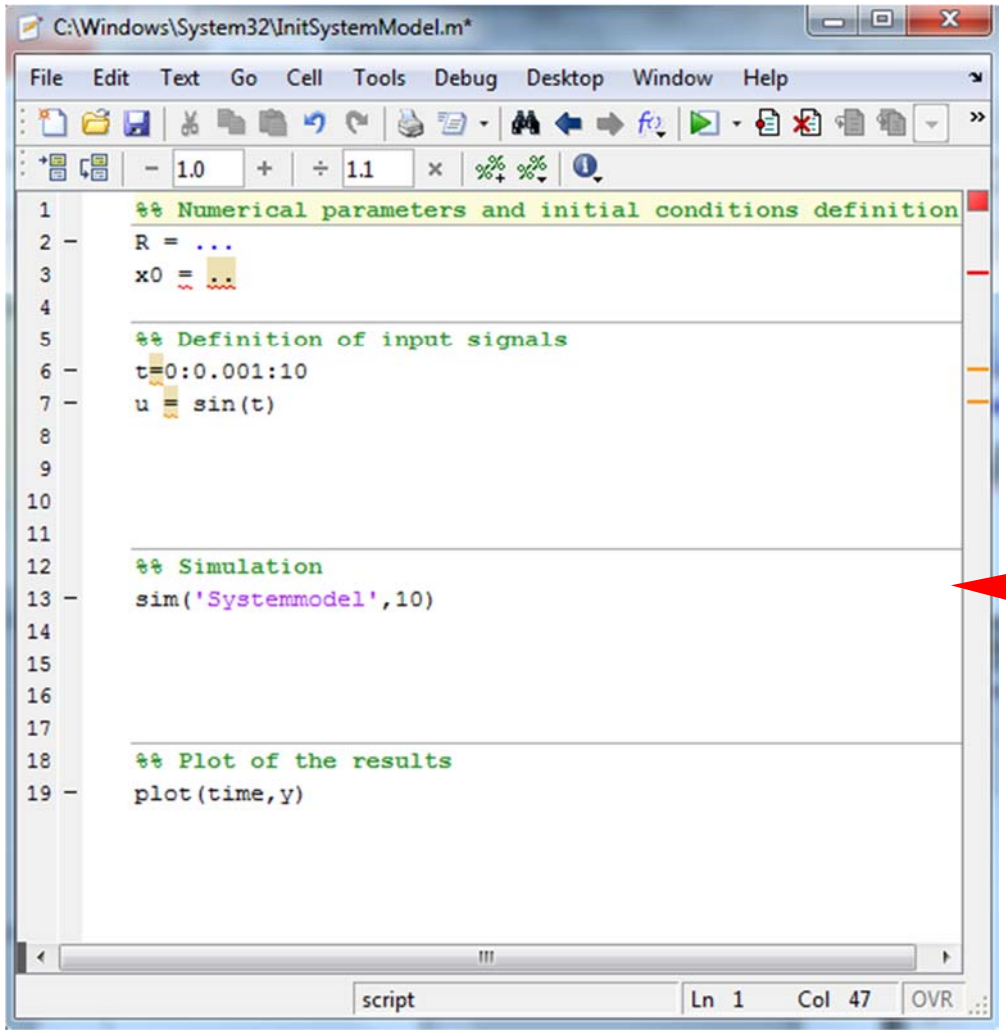
- Display the results

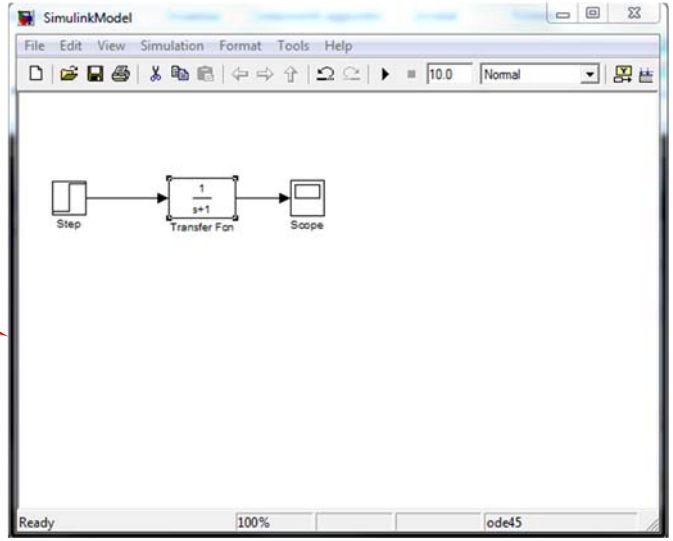
Guidelines for a good simulation

- Matlab-Simulink models are usually structured in two main files:

M-file

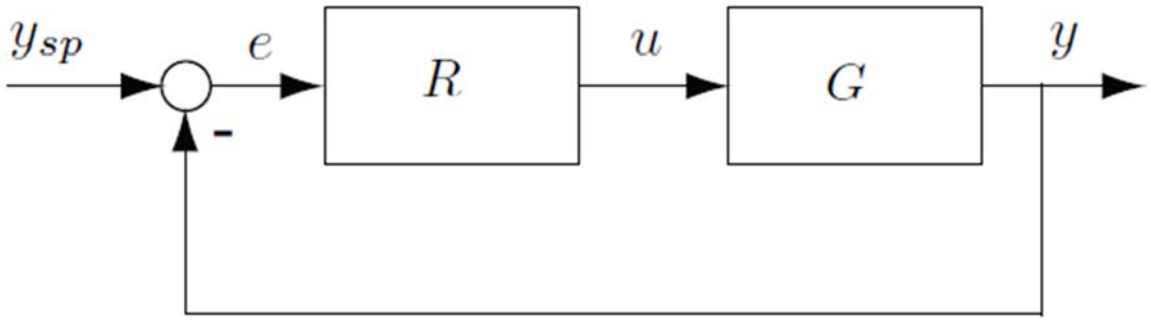
Simulink Model





Simulation of a feedback system - problem

With reference to the scheme of figure



1. Simulate the step response of the system with the continuous-time transfer functions

$$R(s) = \frac{10(s+0.5)}{s} \qquad G(s) = \frac{50}{(s+100)(s+0.5)}$$

In a unique figure plot the system's response $y(t)$ superimposed to the reference signal $y_{sp}(t)$ and the control variable $u(t)$ (two distinct subplots)

Simulation of a feedback system - problem

2. Simulate the step response of the feedback system with the discrete-time transfer functions

$$R(z) = \frac{10 - 9z^{-1}}{1 - z^{-1}}$$

and $G(z)$ obtained from $G(s)$ by discretization with sampling time Ts = 0.1 s. Make the same plots of point 1.

3. Simulate the step response of the feedback system with the discrete-time transfer functions $R(z)$ and the continuous-time plant $G(s)$. Make the same plots of point 1.

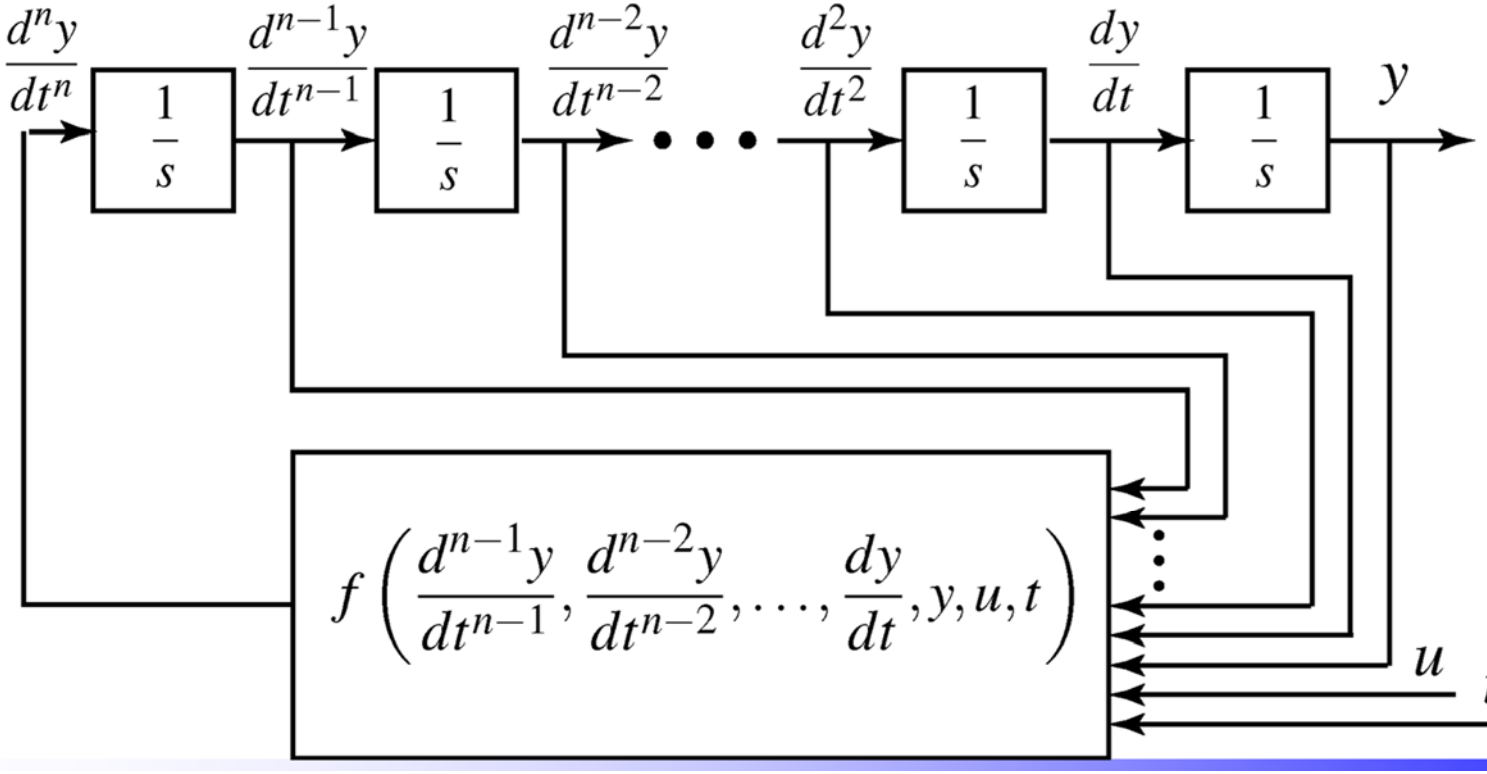
4. Simulate the response of the system considered in the previous point, by considering a reference signal computed with the MATLAB function **`[q_t] = TrjPoly3(q0,q1,T,dt)`**, with **`q0=0, q1=1, T=2, dt =`** Ts.

Simulation of a differential equation

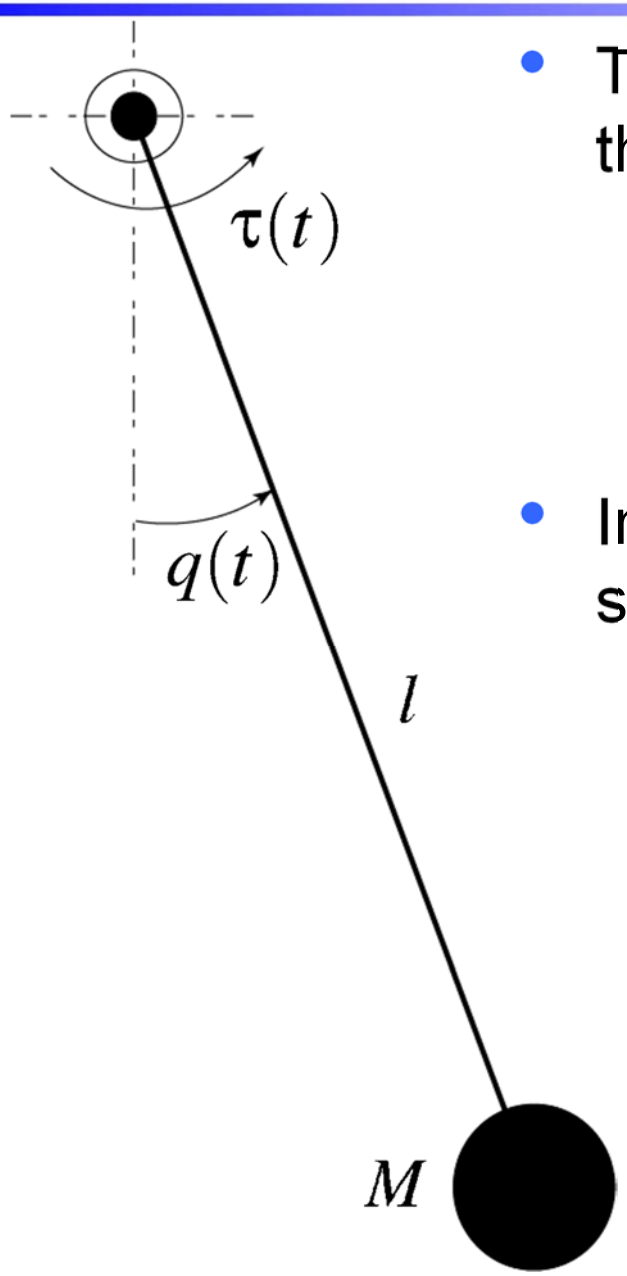
- A Simulink scheme allows the simulation of any differential equation, even nonlinear or time-varying, if it is possible to rewrite it an explicit form, i.e.

$$\frac{d^n y(t)}{dt^n} = f\left(\frac{d^{n-1}y}{dt^{n-1}}, \frac{d^{n-2}y}{dt^{n-2}}, \ldots, \frac{dy}{dt}, y, u, t\right)$$

- The correpsonding block-scheme (to be designed in Simulink) is



Simulation of the simple pendulum



- The dynamics of the system is described by the nonlinear differential equation

$$M l^2 \ddot{q}(t) + M g l \sin(q(t)) = \tau(t)$$

- In order to represent the equation with block-schemes, it is convenient to rewrite it as

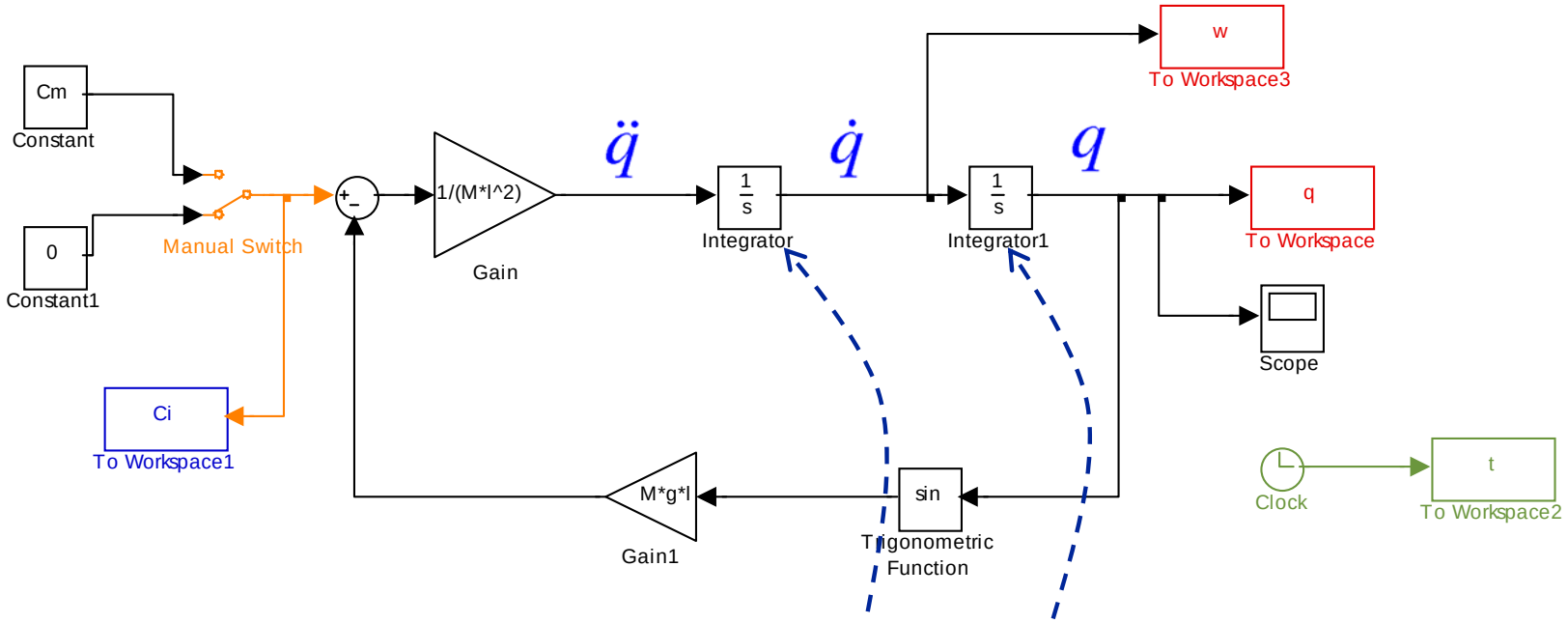
$$\ddot{q}(t) = \frac{1}{M l^2}\left(\tau(t) - M g l \sin(q(t))\right)$$

Simple pendulum – Simulink scheme

- From the equation

$$\ddot{q}(t) = \frac{1}{Ml^2}\left(\tau(t) - Mgl\sin(q(t))\right)$$

the Simulink scheme descends



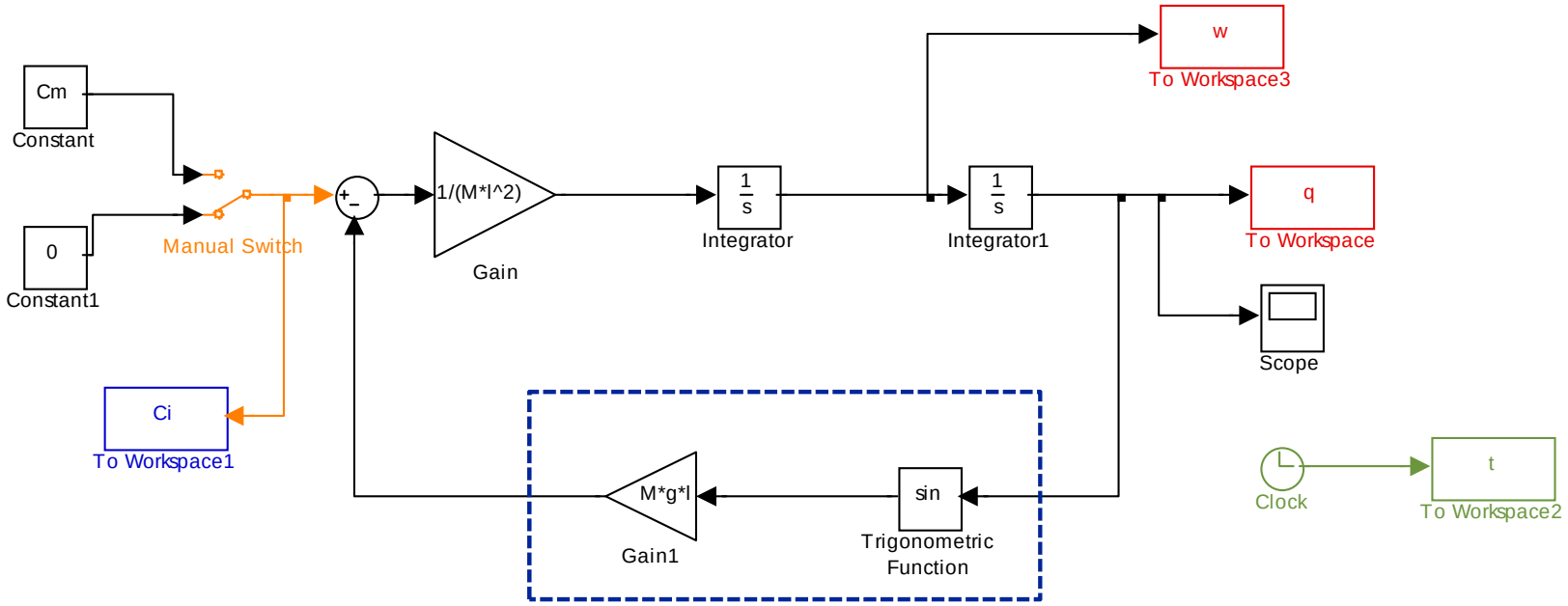
Parameters and initial conditions on $\dot{q}$ and q must be defined in a separate m-file

Simple pendulum – Simulink scheme

- From the equation

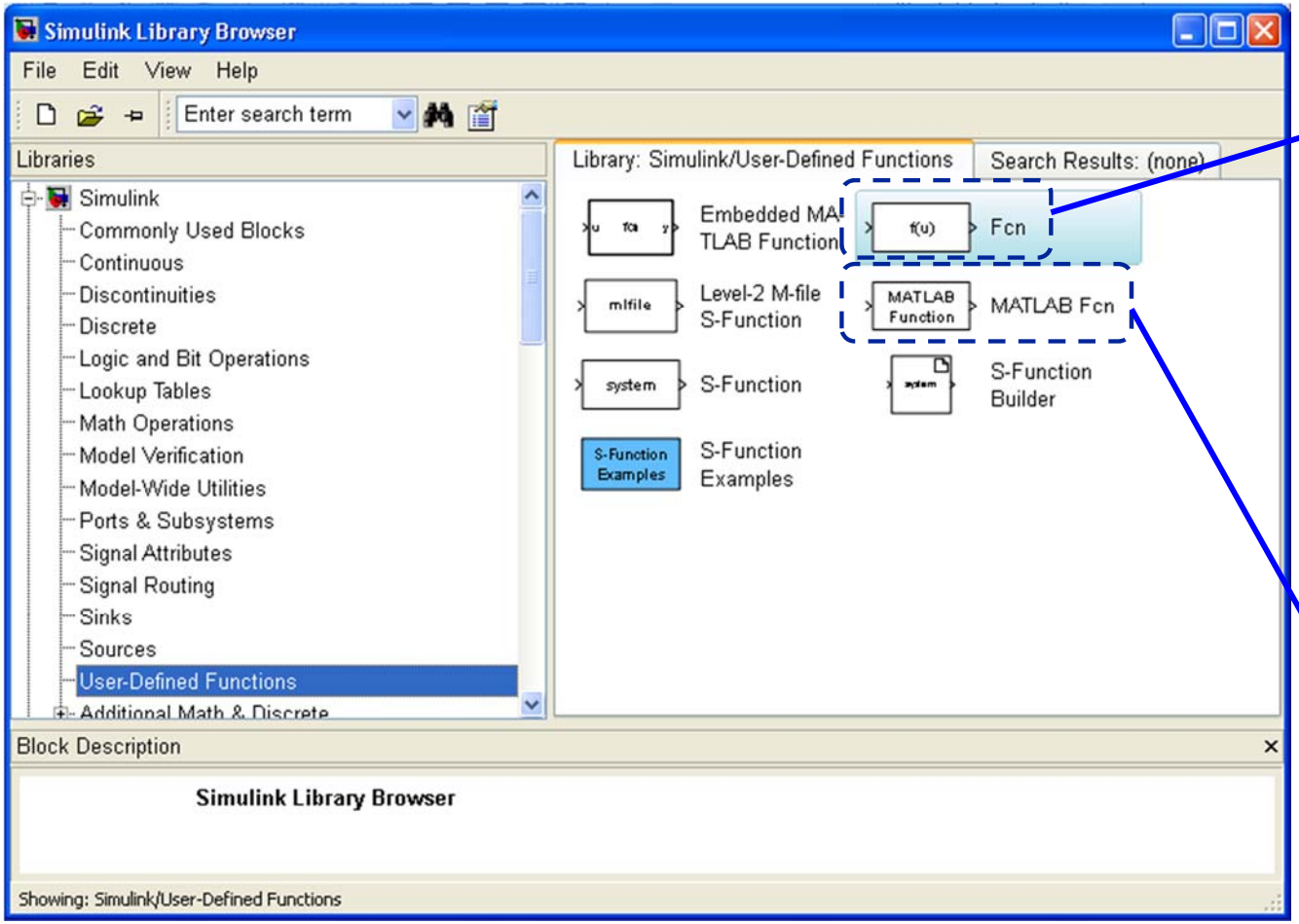
$$\ddot{q}(t) = \frac{1}{Ml^2}\left(\tau(t) - Mgl\sin(q(t))\right)$$

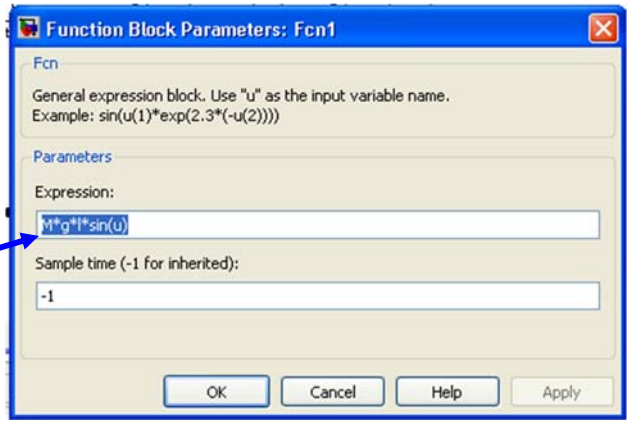
the Simulink scheme descends



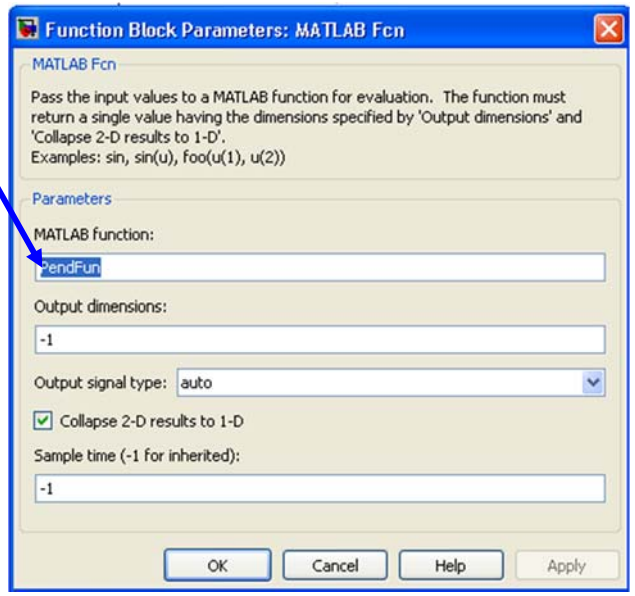
In order to define the function $Mgl\sin(q)$ it is possible to use other Simulink blocks, in particular the blocks that allow the insertion of user-defined functions

Blocks for inserting user-defined functions



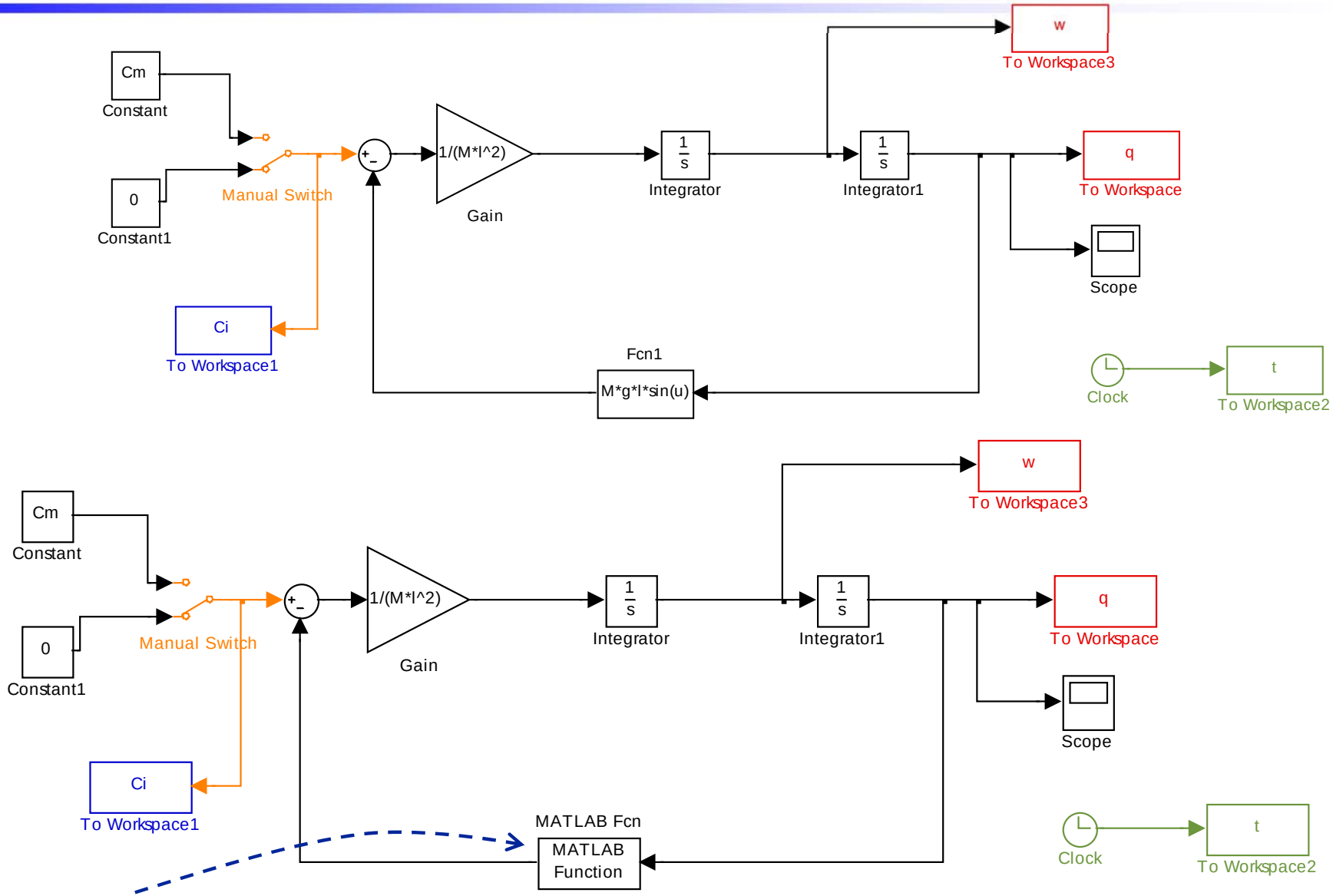


Function directly defined in the Simulink block



External function defined as m-function

Simple pendulum – alternatives schemes




```
function [Out] = PendFun(q)
global M g l;
Out = M*g*l*sin(q);
```

Global parameters in order to make them visible to the function (the same statement is present in the m-file where the parameters are assigned)

Simple pendulum - problem

- Design a Simulink system for solving the equation of a simple pendulum with friction, i.e.

$$M l^2 \ddot{q}(t) + b\dot{q}(t) + M g l \sin(q(t)) = \tau(t)$$

- By assuming the parameters' values

  $M = 0.25$ Kg, $l = 1.2$ m, $g = 9.81$ m/sec^2, $b = 0.25$ Nm/rad sec

  simulate
  - the free response from initial conditions $q(0) = \pi/2, \dot{q}(0) = 0$
  - the forced response to a constant input $\tau(t) = 2$
  - the complete response of the system

Plot in a unique figure (3 distinct subplots) the evolution of in the three cases.

Systems and Control Theory

Master Degree Course in ELECTRONICS ENGINEERING

http://www.dii.unimore.it/~lbiagiotti/SystemsControlTheory.html

Introduction to SIMULINK

Luigi Biagiotti

e-mail: luigi.biagiotti@unimore.it

http://www.dii.unimore.it/~lbiagiotti