

# **SIMULINK FOR BEGINNERS\***

**Heikki Koivo**

\*These notes are written using MATLAB version 7.0 and SIMULINK 6.0. Compared with earlier versions some minor modifications have occurred.

# SIMULINK FOR BEGINNERS:

- To begin your *Simulink* session, start by clicking MATLAB ICON

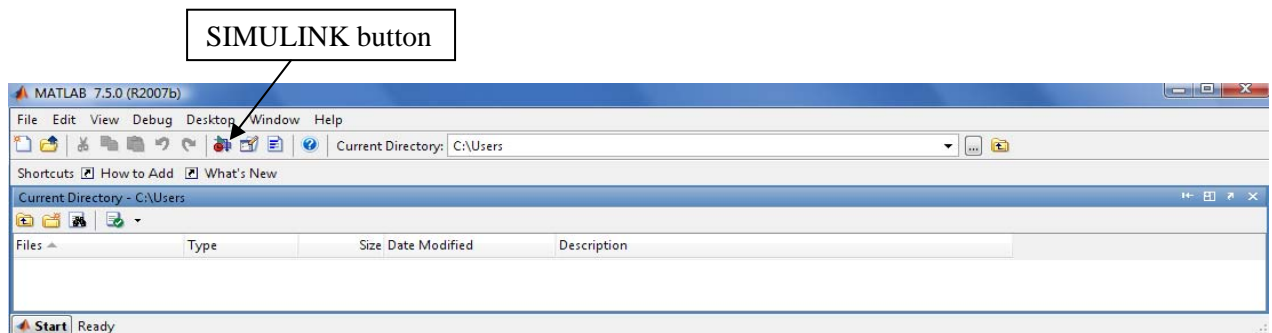


MATLAB 7.0.lnk

twice and then type

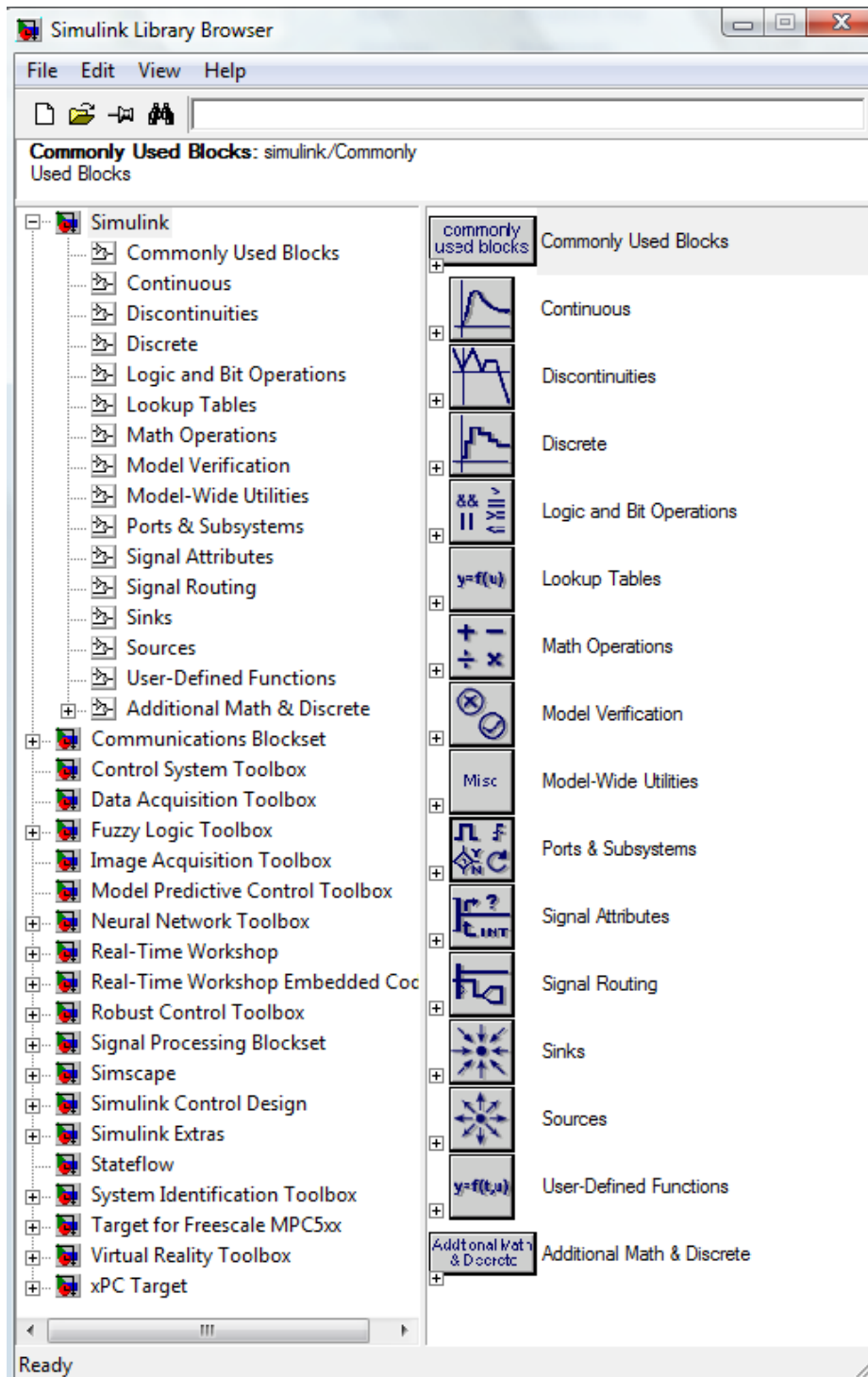
*simulink*

or click *Simulink* button (Figure 1.1).



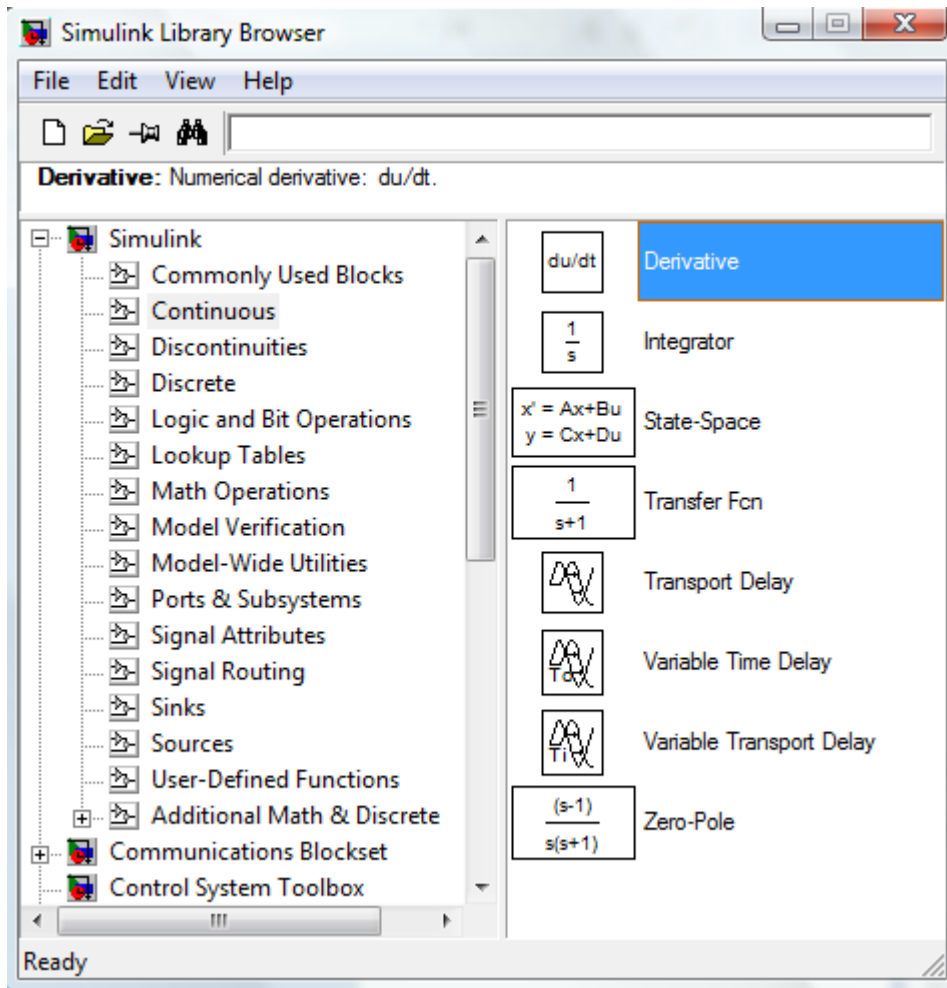
**Figure 1.1** MATLAB window is first opened. To open a new Simulink session either type *simulink* or click the Simulink button.

You will now see the whole *Simulink* block library as in Fig.1.2.



**Figure 1. 2** *Simulink* consists of block libraries *Continuous*, *Discontinuities*, *Discrete*, etc, as shown above. Note that there is a separate *Simulink Extras*, where you can find special blocks.

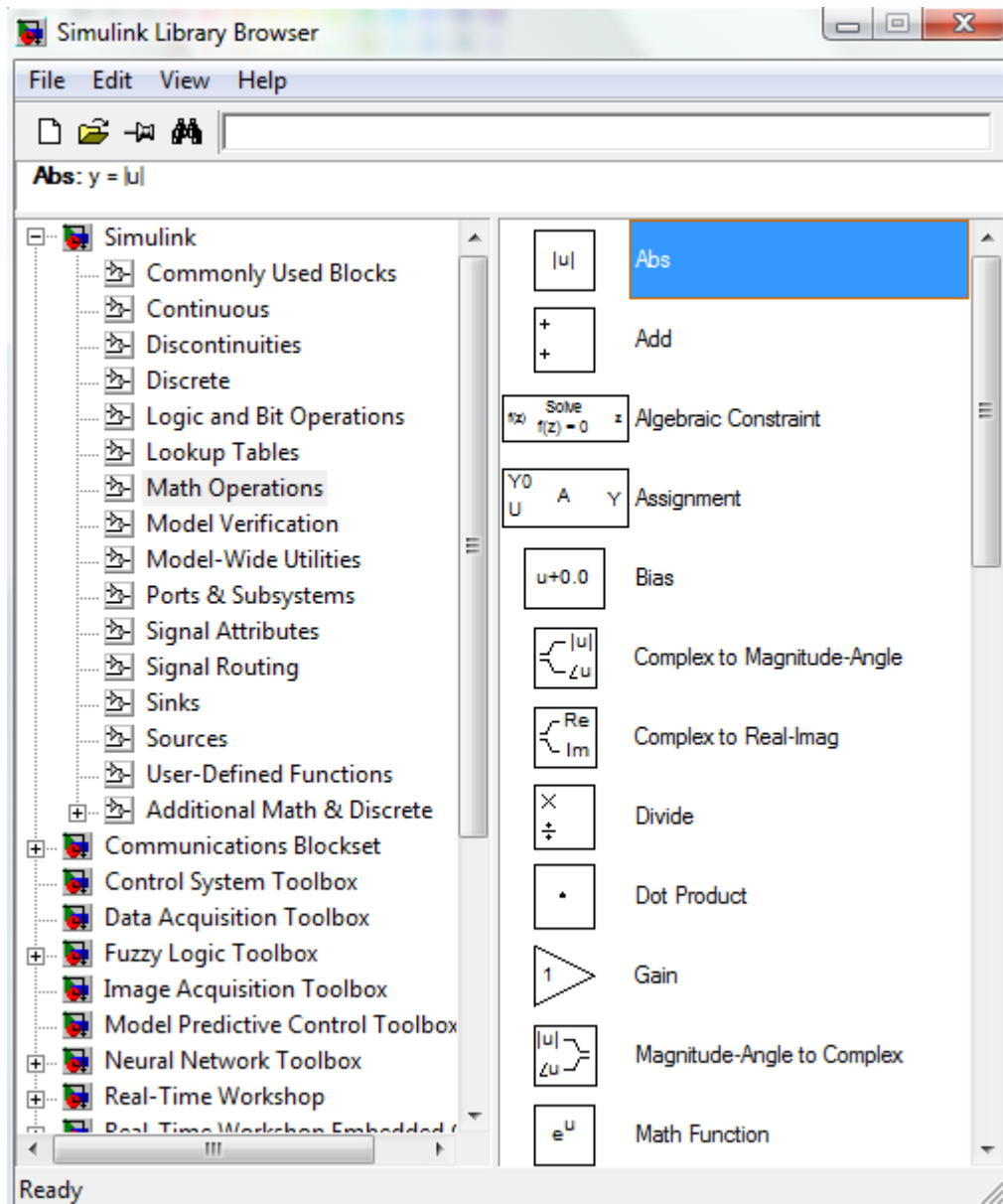
- Browse through different block libraries. E.g., if you click *Continuous*, you will see the following:



**Figure 1.3** Blocks of *Continuous* block library.

Observe the description inside the *Integrator* block  $\frac{1}{s}$ . This operator notation comes from linear systems domain, where  $s$  represents Laplace variable. Roughly speaking,  $s$  corresponds to derivative operator  $\frac{d}{dt}$ , and its inverse  $\frac{1}{s}$ , to integration operation  $\int$ . We will discuss the differences later.

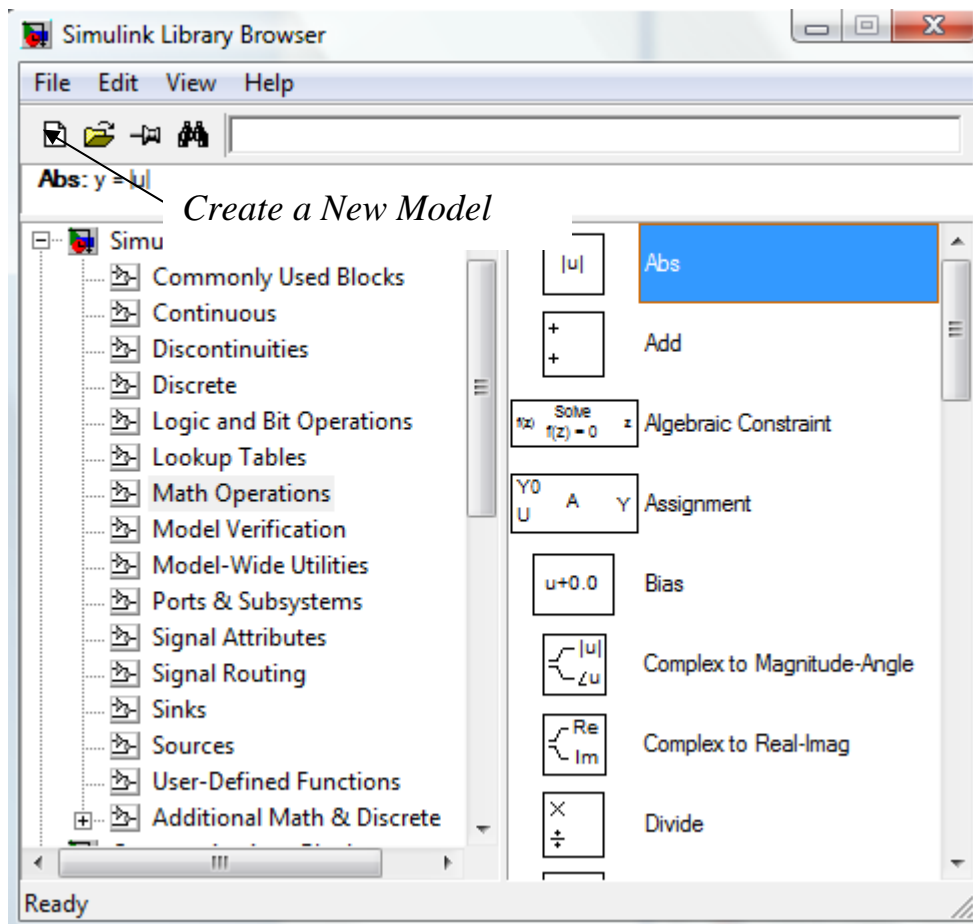
- Math Operations library is shown next



**Figure 1.4** Blocks in Math Operations block library (not all shown).

After browsing through other block libraries, we are now ready to start generating *Simulink* diagrams.

- Choose *File* in menu selection, then *New* and *Model*. This file is called **untitled**. Alternatively, you can click *Create a New Model* icon just below *File* (see Figure 1.5).



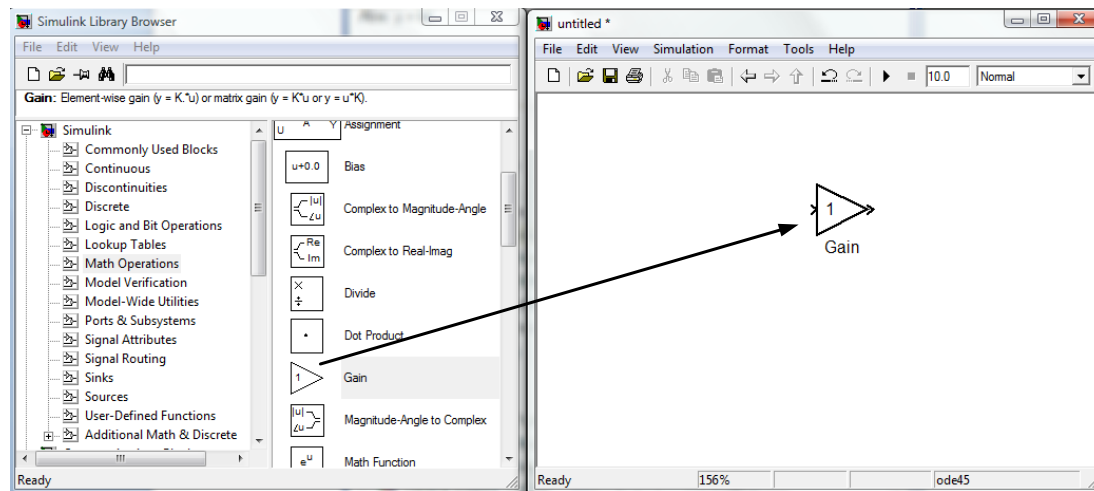
**Figure 1.5** The *Create New Model* button is shown above

You can now construct a *Simulink* configuration of your system using different blocks in the library in the **untitled** file. You do not need to know how to solve e.g. a differential equation, *Simulink* will do it for you. After solving the problem the result is in numerical form, which can also be presented graphically.

**EXAMPLE 1:** Given an input signal  $u(t) = \sin(t)$ . It is fed into an amplifier with gain 2. Simulate the output of the amplifier  $y(t)$ . (The answer is  $y(t) = 2\sin(t)$ )

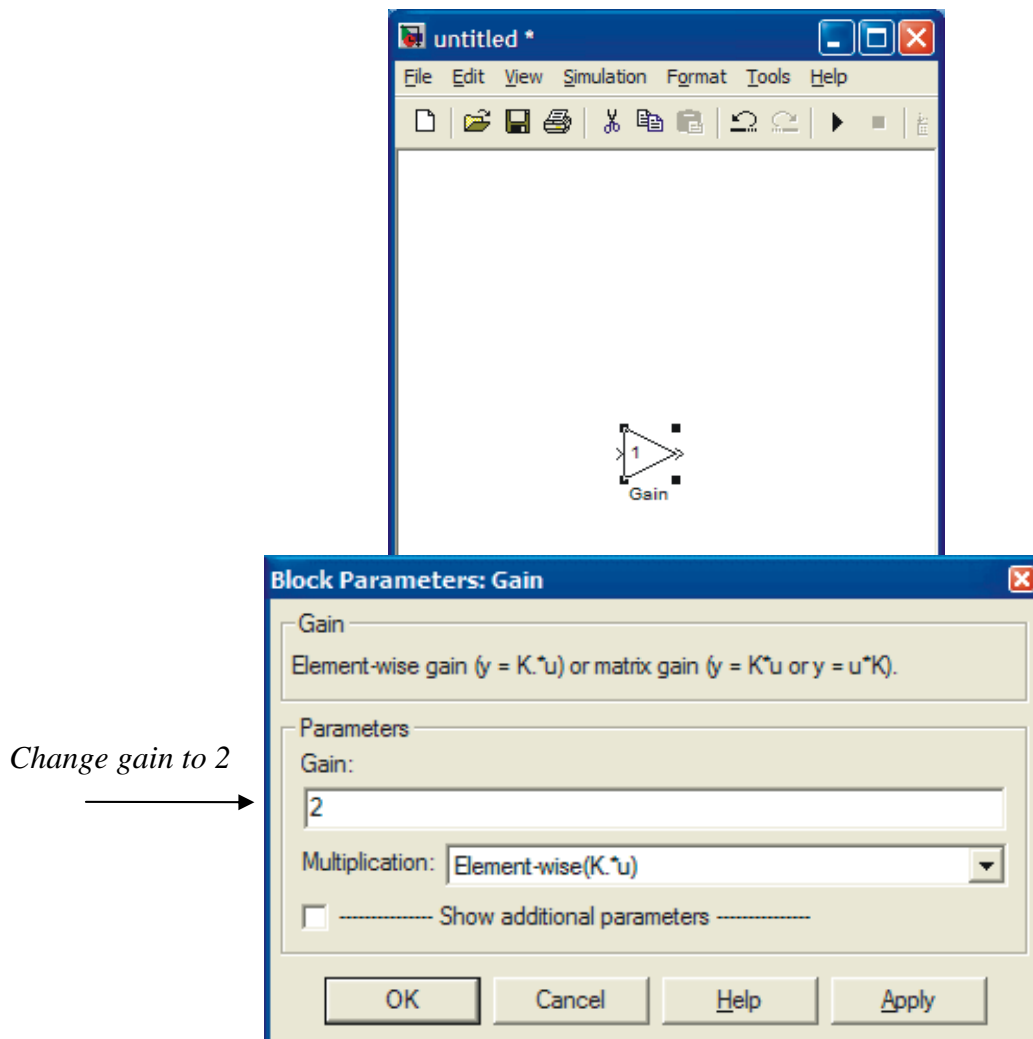
*Solution:* First choose the amplifier block. This you can find under *Math Operations* and it is called *Gain* (Figure 1.6). Move the cursor on top of *Gain*. Press the left hand side of your mouse down and

drag the *Gain* block to the **untitled** file and release it. If you fail, try again. The result is shown below.



**Figure 1. 6** *Gain* block in *Math Operations* block library.

The *Gain* has only one parameter, default value of 1.



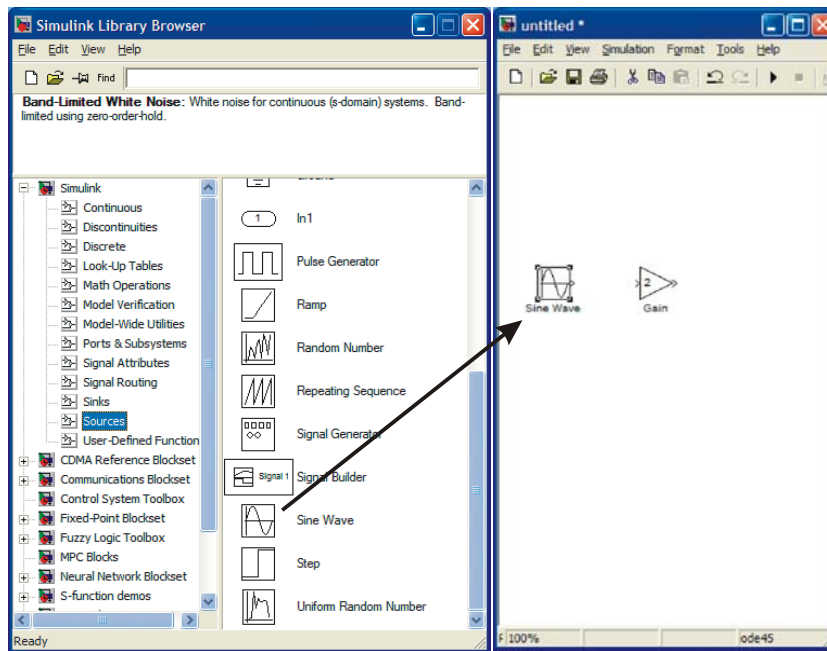
**Figure 1.7** Changing the gain value in the *Gain* block from 1 to 2.

The gain value can be changed by moving the cursor over the *Gain* block. Open the block by clicking it twice and go to Gain where you see 1. Change it to 2 (Figure 1. 8). Once finished, click OK. If you do not, the value will not be changed. The numerical value inside the *Gain* block will now become 2.

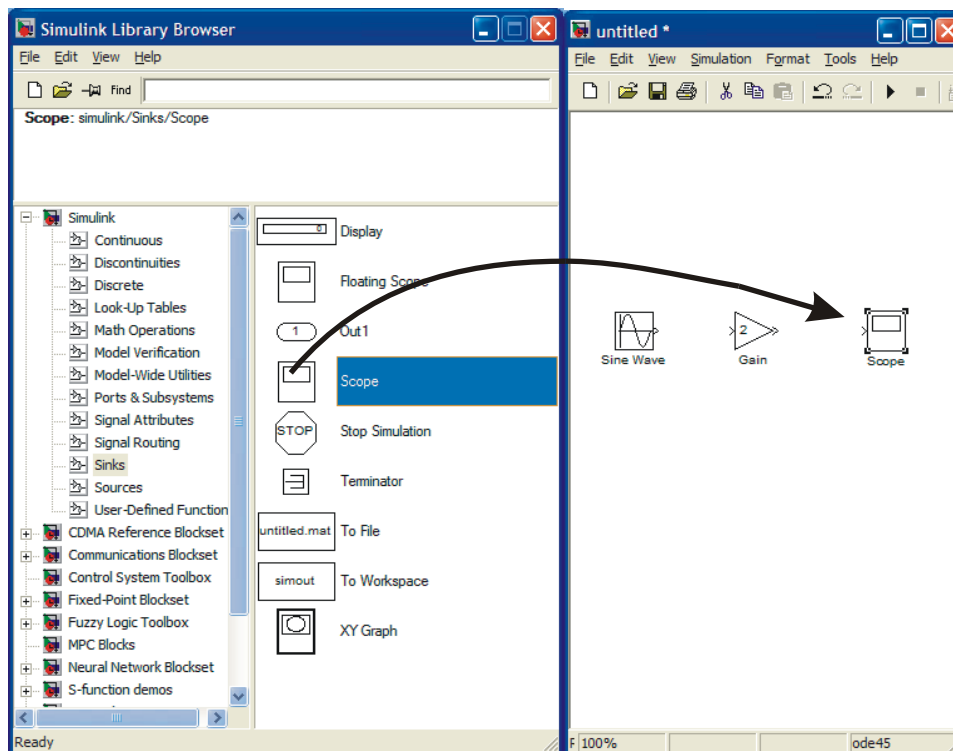
Next find input signals. Input block library is called *Sources*, where you can find the needed input: *Sine Wave*. Move the cursor on top of the *Sine Wave* block, drag the block to the **untitled** file and then release it (Figure 1. 9).

To see the result, you need to install a *Sink* from *Sinks* library. In the beginning, the easiest *Sink* device is *Scope*. Move the *Scope* block to the **untitled** file in the same way as before (The result is shown below Figure 1. 10).





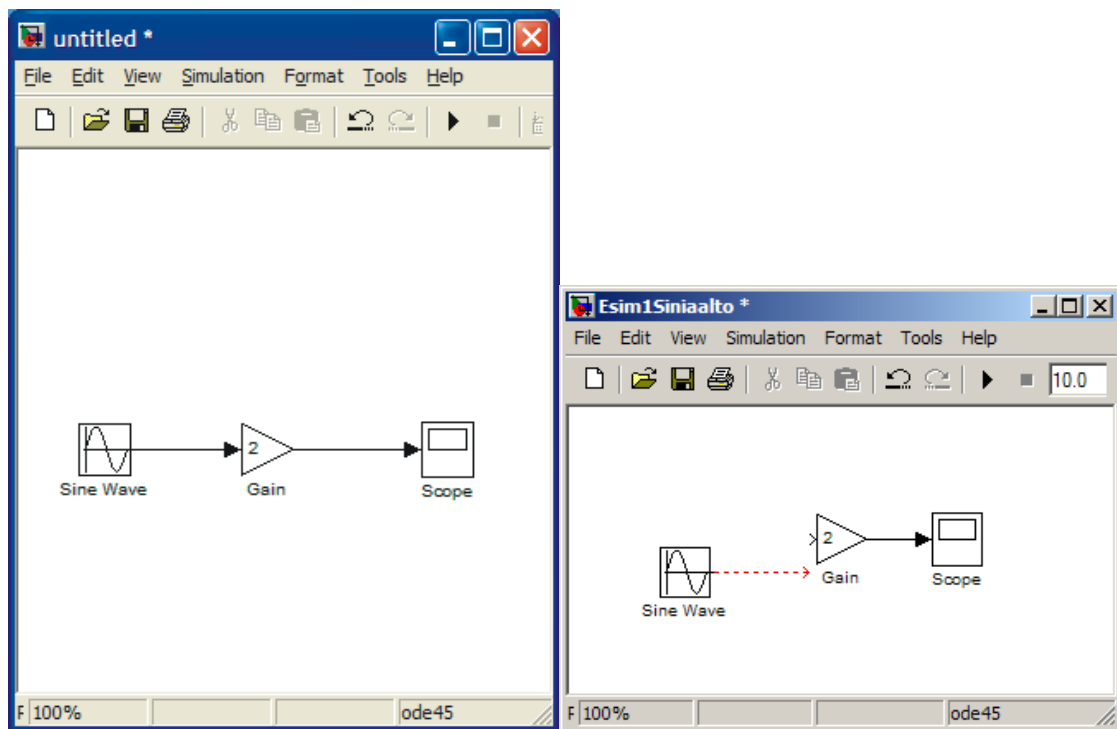
**Figure 1.9** *Sine Wave* block is taken from the *Sources* block library.



**Figure 1.10** *Scope* block is taken from the *Sources* block library.

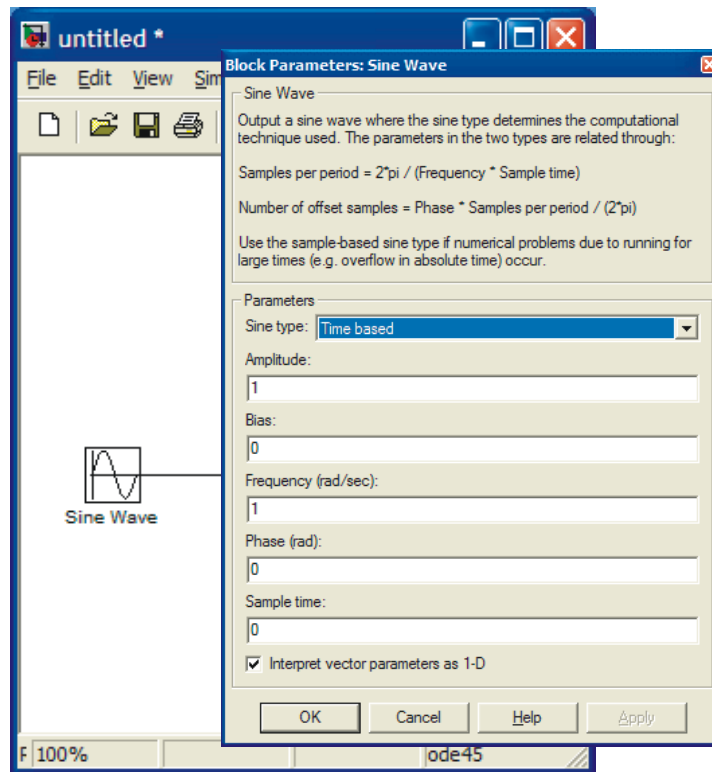
The only thing missing of the system is to connect the subsystems together. This is done with the mouse. Take the cursor to the output of *Sine Wave* block. You'll see a hairline cursor, when you are close enough. Now press the mouse button down. Keep it down and move it close to the input of the *Gain* block. While you drag your

mouse, you will see a dashed black line forming from the output side of the block which you started. Once you reach the input of the other block, a solid black line is formed to indicate that connection is made. If the connection is not complete, you will see a (red) dotted line (Figure 1.11). To complete the connection, you can e.g. activate the receiving block with your cursor and drag the block so that its input will meet the arrowhead.



**Figure 1.11** Figure on the left shows the completely connected system configuration. Figure on the right shows an incomplete connection with a (red) dashed line.

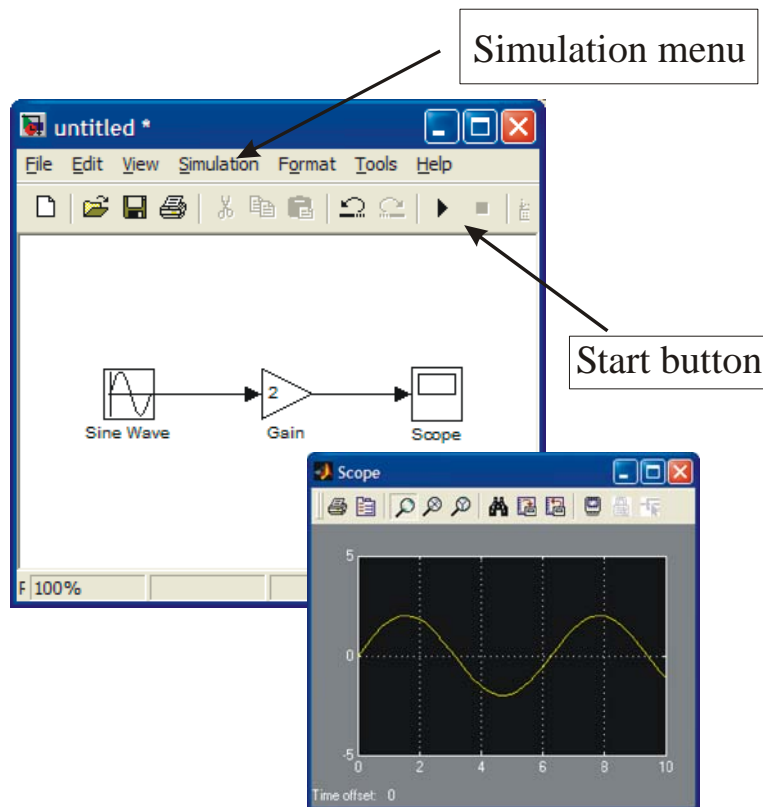
Your configuration of the system is now complete. Before simulation you should check that the parameter values in the *Sine Wave* block are correct. Open it by placing the cursor on top of the block and click twice.



**Figure 1. 12** *Block Parameters* of the *Sine Wave* are shown on the right.

For this example *Amplitude* is 1, *Frequency* (rad/s) is also 1, *Phase* and *Bias* are 0, so default values are OK. Sampling is not an issue here.

You can begin simulation by choosing *Start* from Simulation menu or by clicking the start button (Figure 1.13).



**Figure 1.13** Starting simulation either with the *Start* button or from *Simulation* menu. The simulation result is shown in the scope. The amplitude should be equal to 2 as seen in the Figure.

The solution is seen in the *Scope*. If the *Scope* does not open automatically, click it twice. Observe from the *x*-axis that the default simulation time is 10 s. If the figure is not scaled properly, use the different scale buttons in the Menu above the figure. For example, the button looking like binoculars performs autoscaling of the figure.

**An important thing in learning to use SIMULINK is to actively play and experiment!**

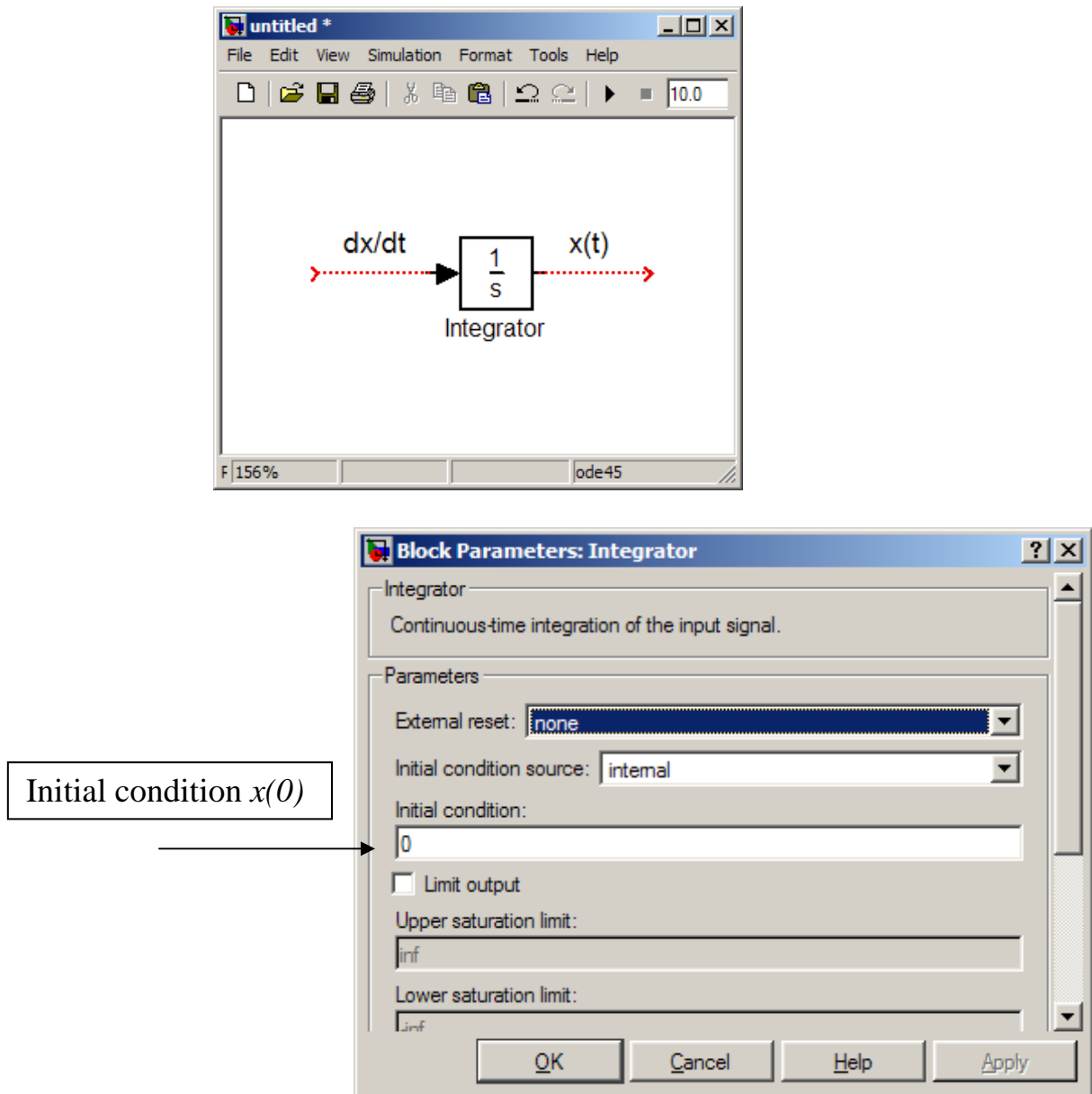
**Example 2 – First order differential equation:** Solve numerically the differential equation with SIMULINK. Think also what the exact solution is.

$$\begin{aligned}\frac{dx}{dt} &= -2x + 1, \quad t > 0 \\ x(0) &= 0\end{aligned}\tag{1.1}$$

**Solution 1:** Consider the integrator block in *Continuous* block library. The input to the block is  $\frac{dx}{dt}$ . The integrator integrates it from  $t=0$  s up to  $t$ . Default value is  $t=10$  s.

$$\int_0^t \frac{dx}{dt} dt = x(t) - x(0)\tag{1.2}$$

What is shown is only the input  $\frac{dx}{dt}$  and output  $x(t)$ . The initial condition is given in the *Block parameters* (Figure 1.14).

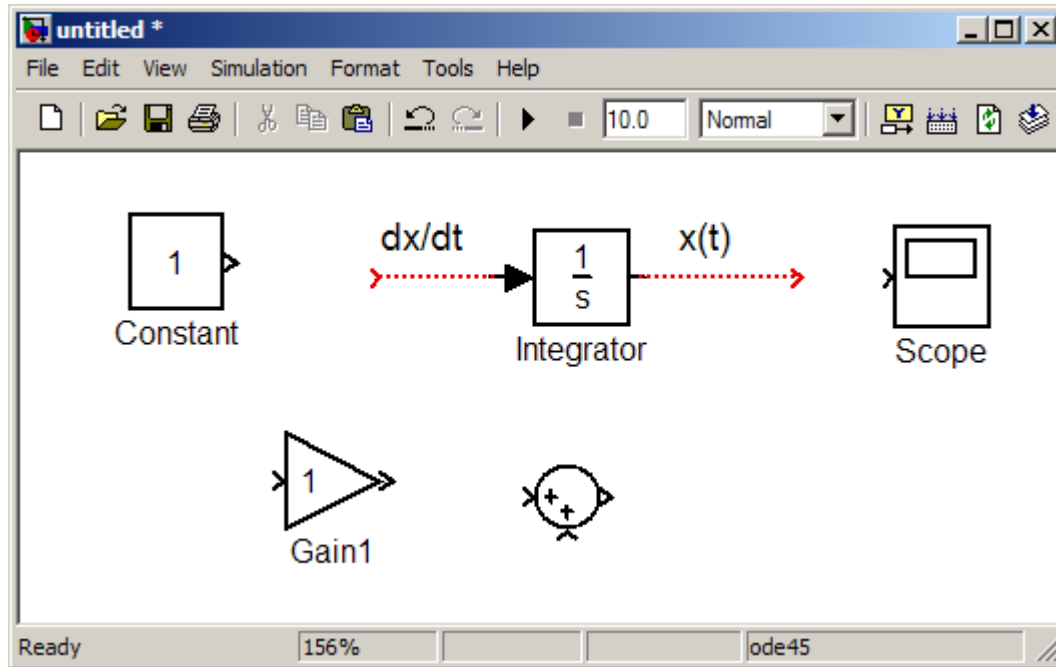


**Figure 1.14** Operation of the integrator block: Input is  $\frac{dx}{dt}$  and output  $x(t)$  (above). Initial condition  $x(0)$  is given by clicking the block twice revealing the block parameters. Default value is  $x(0)=0$  (below).

Continuing our configuration we observe that in Figure 1.14 we already have the left hand side of equation (1.1), which is  $\frac{dx}{dt}$ . We still need to generate the right hand side of the equation  $-2x+1$  and equate it with  $\frac{dx}{dt}$ .

The integration in Figure 1.14 produces  $x(t)$ . This needs to be multiplied by 2 (or -2), minus sign can also be added later to produce  $-2x$ . What is

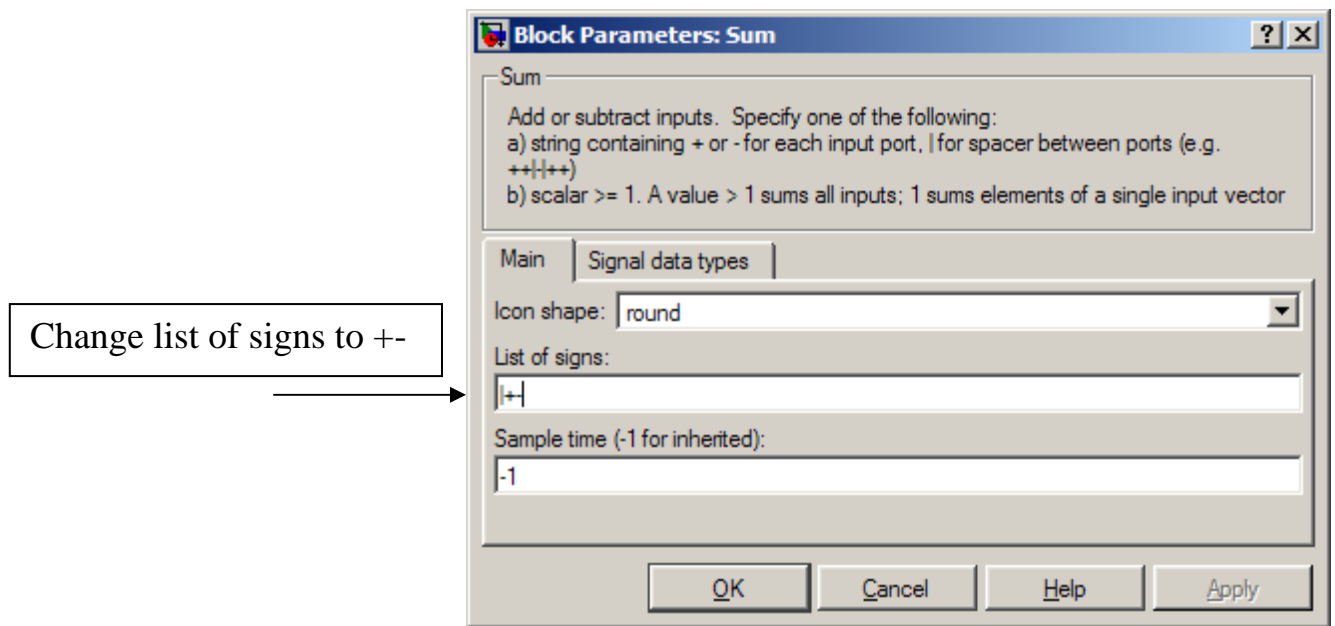
needed is the gain block from *Math operations* block library. Finally, a constant from the *Sources* block library is picked. For summing we need a summing block, which is found from *Math operations* block library. All the mentioned blocks are shown in Figure 1.15.



**Figure 1.15** Blocks needed to configure system (1.1).

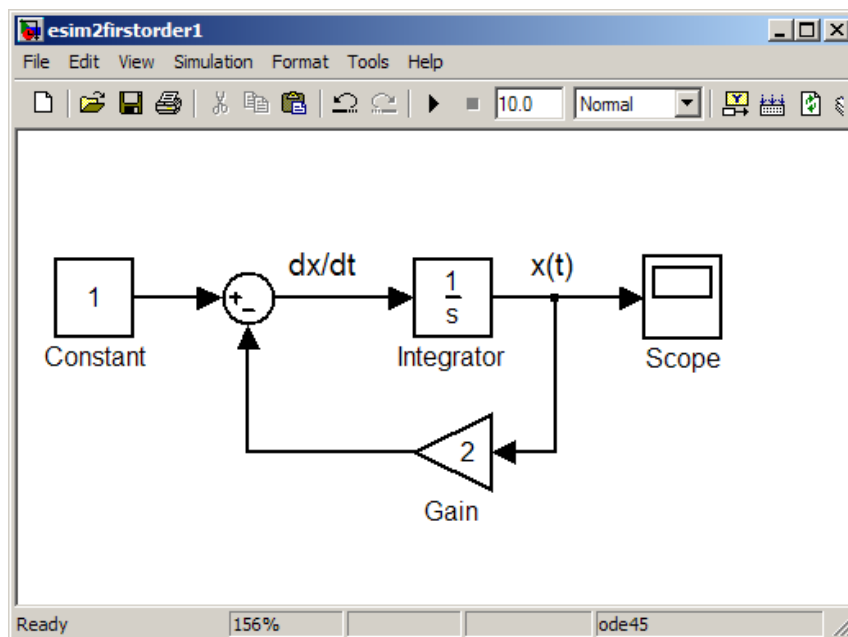
Activate *Gain* block by clicking once and apply *Flip block*, which is found under *Format*. Open the *Gain* block to change the gain value to 2. Remember to click OK. With the *Sum* block, click it open and change one sign to a minus sign. Again remember to click OK.

Default value for *Integrator's Initial condition*  $x(0)$  is 0, so we do not need to change it.



**Figure 1.16** Block parameters of *Sum*. Change list of signs to +-.

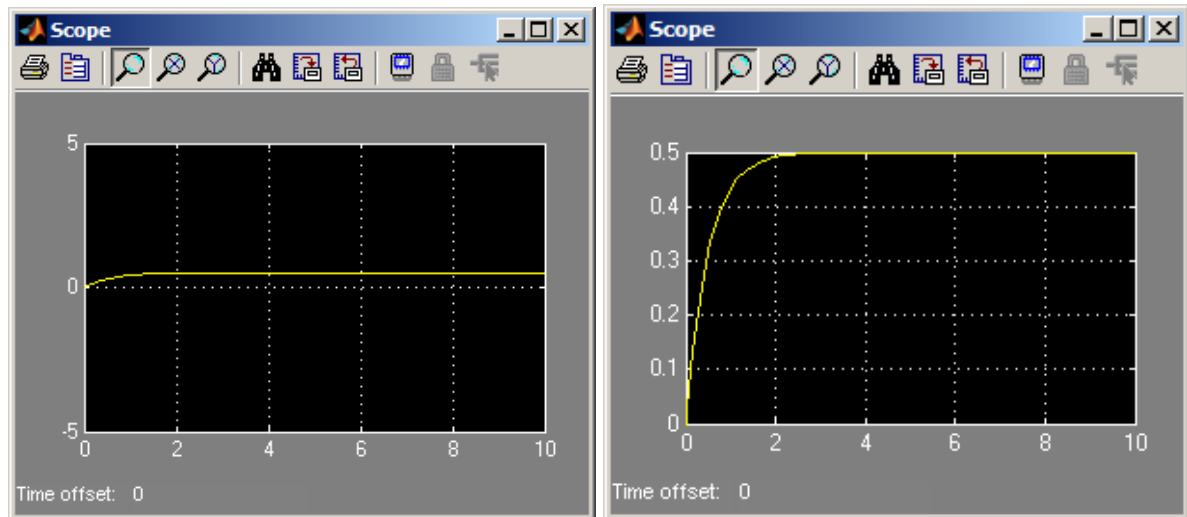
The final configuration is given in Figure 1.17. *Scope* has been added to observe the simulation result.



**Figure 1.17** The final SIMULINK configuration of system (1.1).

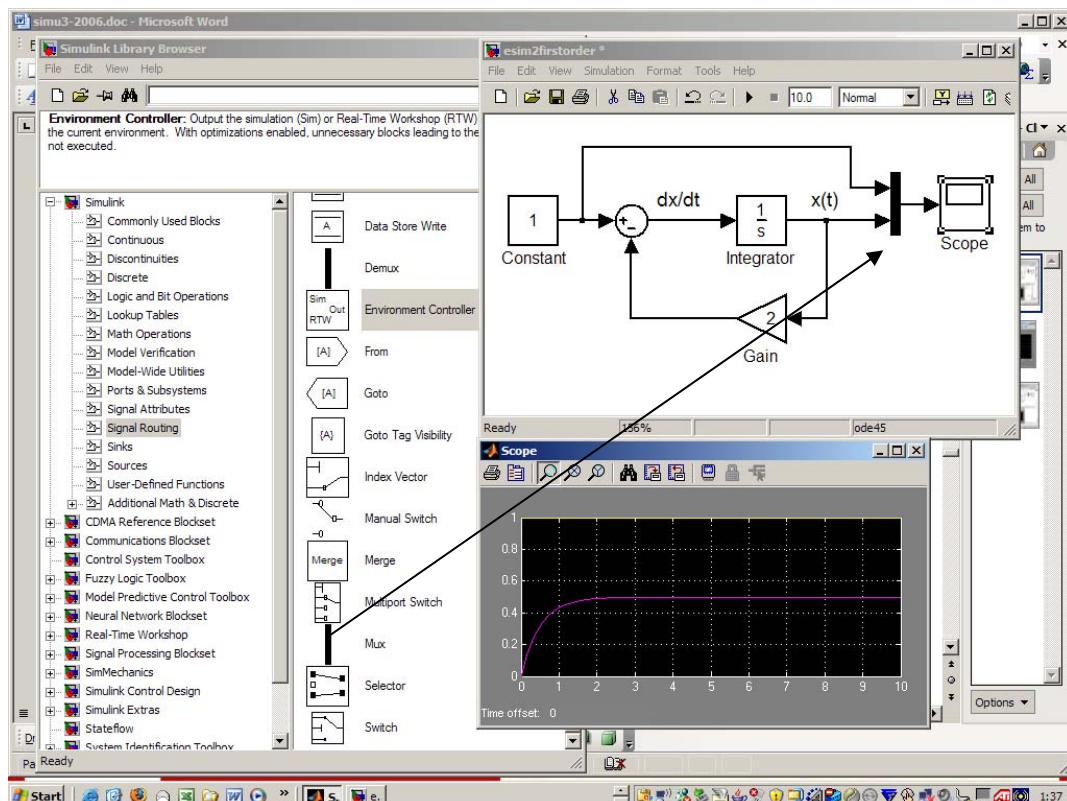
Start the simulation with either the *Simulation* button or *Start* from the *Simulation* menu. The result is shown in Figure 1.18.





**Figure 1.18** The simulation result on the left. Same result on the right after *Autoscaling*.

The result is a typical exponential response of a forced first order linear system. If you want to see both input and output at the same time in the same figure, use *Mux* (multiplexer) block, which you can find in the *Signals Routing* block library. Configure the system as shown in Figure 1.19.



**Figure 1.19** Multiplexer block *Mux* allows simultaneous display of both input and output signals.

**Solution 2 (Transfer function):** Another way to configure system (1.1) is to use a differential operator  $D$  and derive a transfer function for the system. Let  $D = \frac{d}{dt}$  and denote the input in (1.1) as  $u(t)=1$ . Then the differential equation becomes

$$Dx = -2x + u(t) \quad (1.3)$$

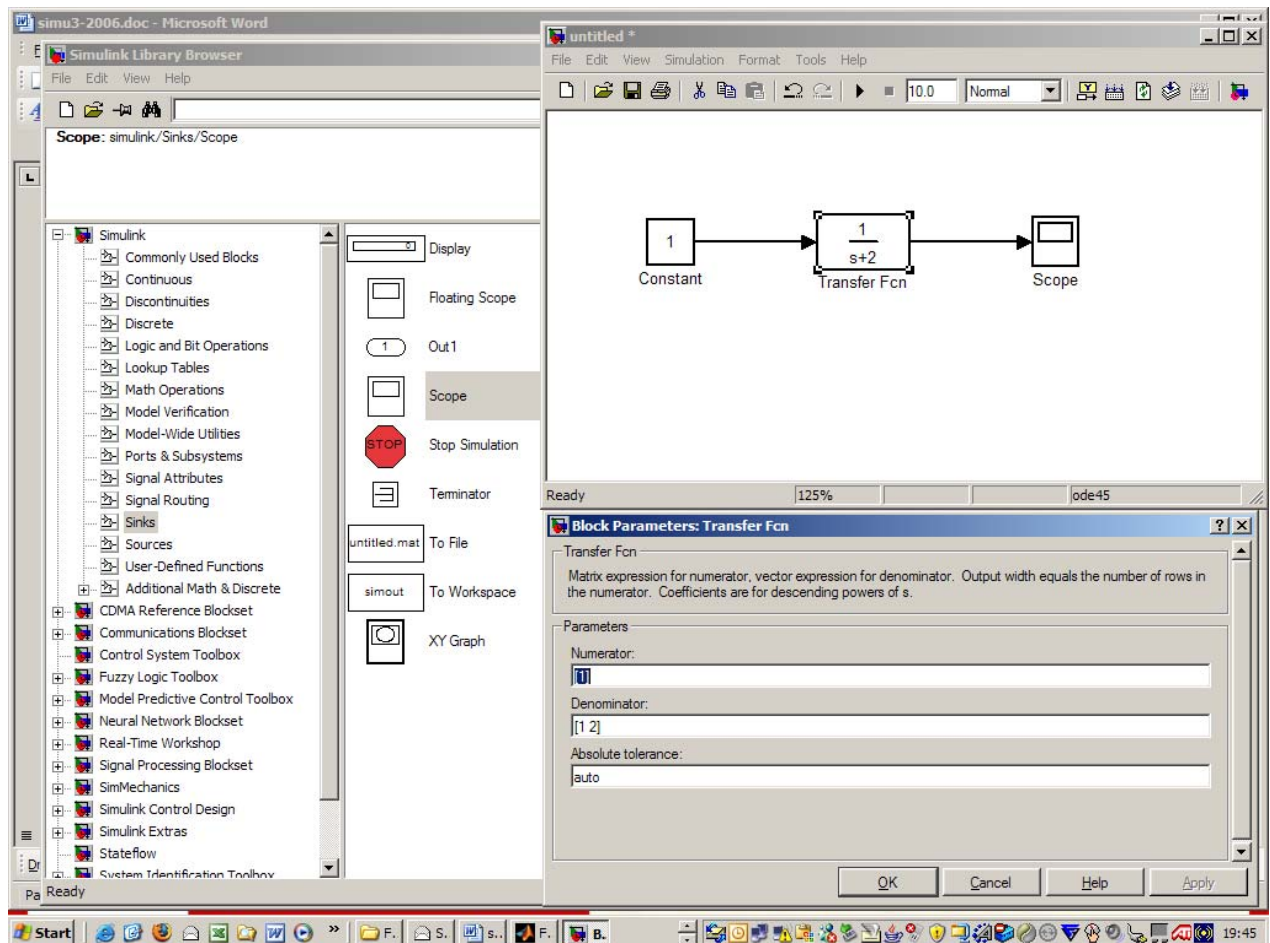
or

$$Dx + 2x = (D + 2)x = u \quad (1.4)$$

Solving for  $x$  leads to

$$\underbrace{\overset{x}{\underbrace{\quad}_{\text{output}}}}_{\text{output}} = \frac{1}{\underbrace{D+2}_{\text{transfer function}}} \underbrace{u}_{\text{input}} \quad (1.5)$$

The configuration is now shown in Figure 1.20, where  $D$  has been replaced by  $s$ . The coefficients in the denominator polynomial are changed from default values  $[1 \ 1]$  to  $[1 \ 2]$ .



**Figure 1. 20** System (1.1) in block diagram form. The Block Parameters in Transfer Function are [1 1] and have to be changed: Denominator multipliers are [2 1]. This system configuration is equivalent to that of Figure 1.17, except here the initial condition is equal to zero.

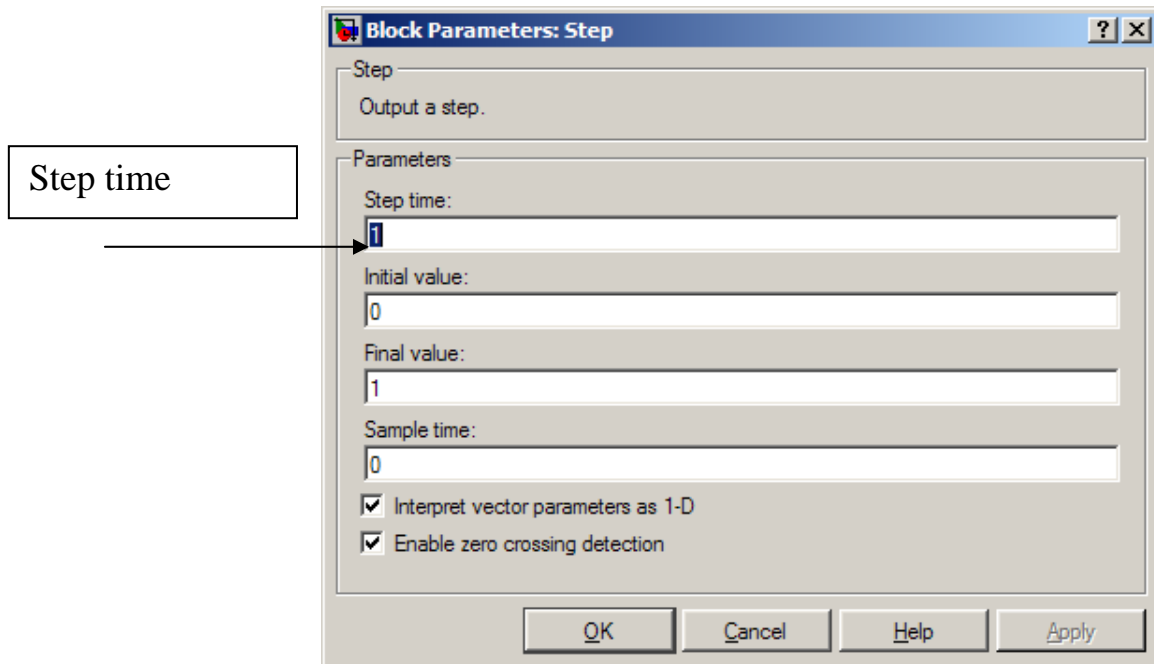
**Step input:** The input in system (1.1) is 1 after  $t \geq 0$ .

A function, that first has a constant value say 0, and then at some later time instant changes to another constant value, is called a *Step function*  $u(t)$ . It is defined as

$$u(t) = \begin{cases} 1, & t > 0 \\ 0, & t < 0 \end{cases} \quad (1.6)$$

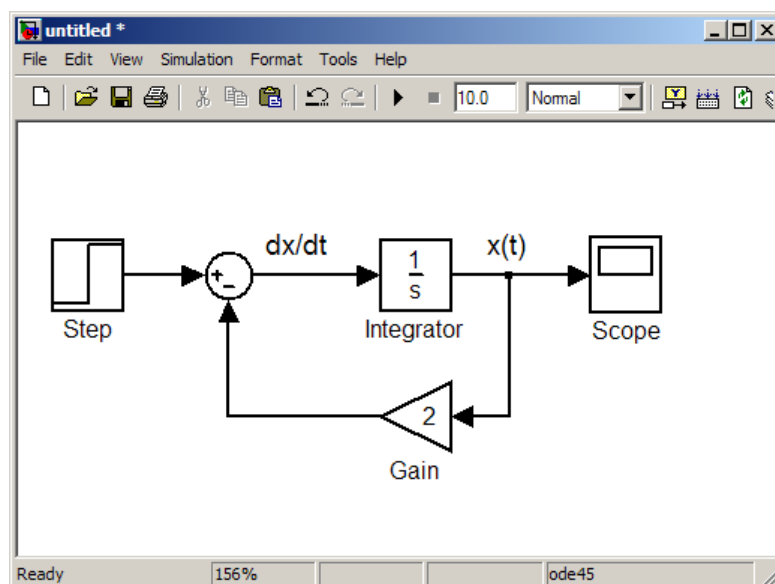
Here the change happens at  $t = 0$  s. Note that it does not matter what value the function assumes at  $t = 0$  s.

*Step function* can be found from the **Sources** block library as *Step* block. Note however that the default stepping time is not  $t=0$  s but  $t=1$  s. This can easily be changed as seen in Figure 1.21.



**Figure 1.21** Changing *Block Parameters* in *Step Function*, especially the *Step Time*.

Instead of using *Constant* block, it is common to use *Step* block as shown in Figure 1.22. The simulation result is exactly the same as in Fig. 1.17, if the change of Figure 1.21 is done.



**Figure 1.22** Configuration of (1.1) with a *Step* block.

## MATLAB Command window

Once you have defined your system in SIMULINK window, you can simulate it also on the MATLAB Command window. Save your model – it has first the name *untitled*, which may be used here. Go to MATLAB command window and type **help sim**. The following lines make you understand how to simulate from Command window.

» help sim

### SIM Simulate a Simulink model

SIM('model') will simulate your Simulink model using all simulation parameter dialog settings including Workspace I/O options.

The SIM command also takes the following parameters. By default time, state, and output are saved to the specified left hand side arguments unless OPTIONS overrides this. If there are no left hand side arguments, then the simulation parameters dialog Workspace I/O settings are used to specify what data to log.

```
[T,X,Y] = SIM('model',TIMESPAN,OPTIONS,UT)
[T,X,Y1,...,Yn] = SIM('model',TIMESPAN,OPTIONS,UT)
```

T : Returned time vector.  
 X : Returned state in matrix or structure format.  
     The state matrix contains continuous states followed by discrete states.  
 Y : Returned output in matrix or structure format.  
     For block diagram models this contains all root-level output blocks.  
 Y1,...,Yn : Can only be specified for block diagram models, where n must be the number of root-level output blocks. Each output will be returned in the Y1,...,Yn variables.

'model' : Name of a block diagram model.

TIMESPAN : One of:

TFinal,  
 [TStart TFinal], or  
 [TStart OutputTimes TFinal].

OutputTimes are time points which will be returned in T, but in general T will include additional time points.

OPTIONS : Optional simulation parameters. This is a structure created with SIMSET using name value pairs.

UT : Optional extern input. UT = [T, U1, ... Un]  
 where T = [t1, ..., tm]' or UT is a string containing a function u=UT(t) evaluated at each time step. For table inputs, the input to the model is interpolated from UT.

Specifying any right hand side argument to `SIM` as the empty matrix, `[]`, will cause the default for the argument to be used.

Only the first parameter is required. All defaults will be taken from the block diagram, including unspecified options. Any optional arguments specified will override the settings in the block diagram.

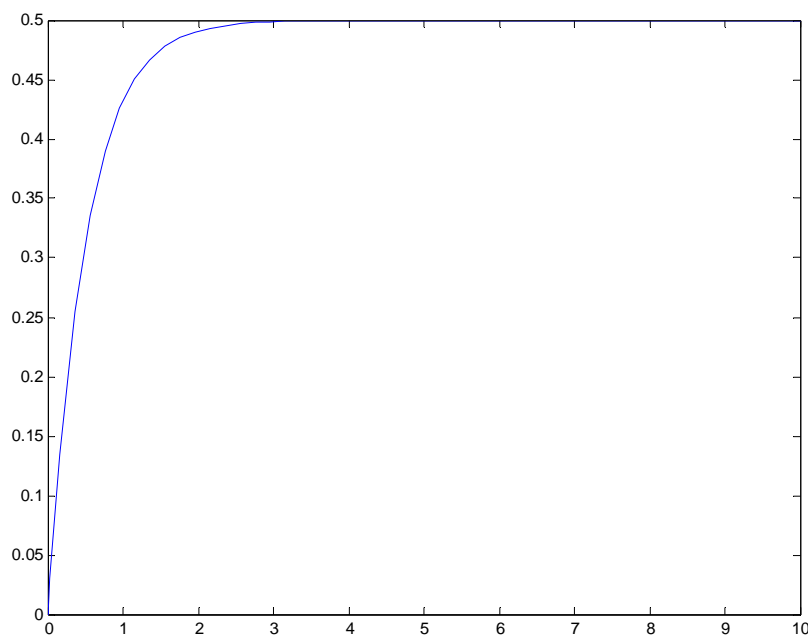
See also `SLDEBUG`, `SIMSET`.  
Overloaded methods  
`help network/sim.m`

The simplest way to start is to use *sim* command in the following way.

```
[t,x,y]=sim('untitled');
```

The only thing needed here is the alphanumerical string indicating the name of the system you want to simulate. Next you can use `plot` command to see the result.

```
plot(t,x)
```

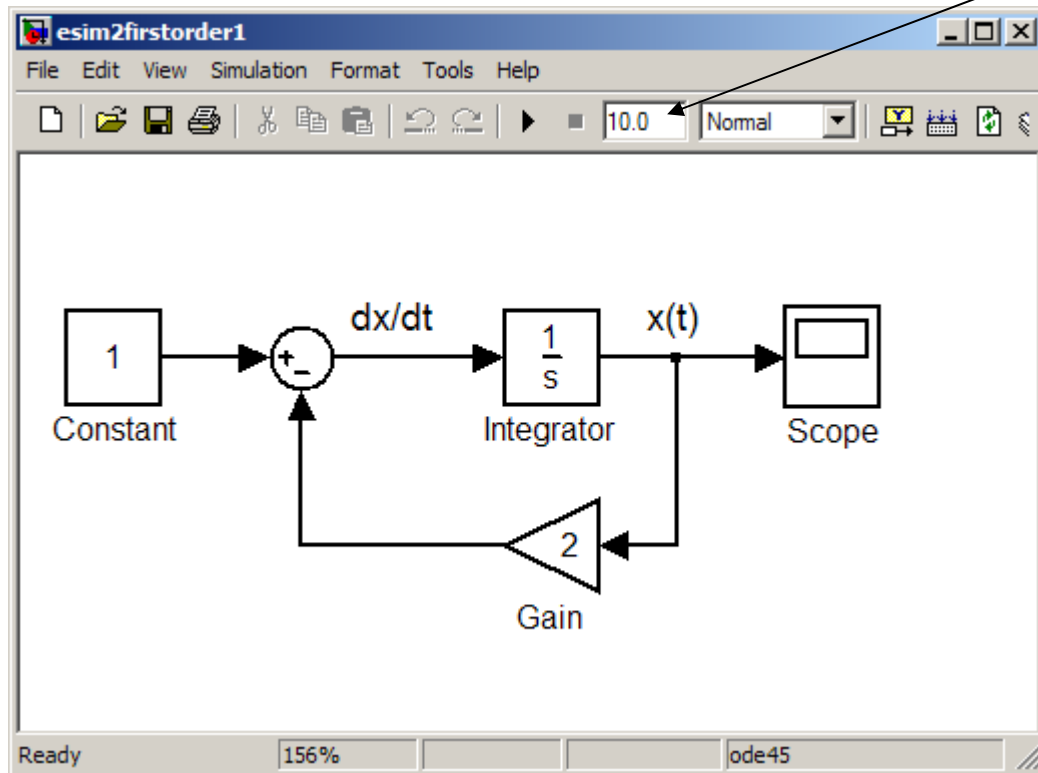


**Figure 1. 23** Simulation result is plotted with *plot* command.

To see *y*, you need to introduce an output in the SIMULINK model.

The default time for simulation is 10 s. If you wish to simulate longer you can change it from *Simulation stop time* (Figure 1.24).

Alternatively, open *Simulation* menu, choose *Configuration parameters*.



**Figure 1. 24** Simulation stop time can be changed from 10 seconds to a larger or smaller nonnegative value.

Complete the exercise. Run the simulation from SIMULINK and also from Command window.

### Example 1.3 Damped oscillator

Solve the damped oscillator problem

$$\frac{d^2x}{dt^2} + 5\frac{dx}{dt} + 9x = 0 \quad (1.7)$$

$$\begin{aligned}\frac{dx}{dt} &= \dot{x}(0) = -2 \\ x(0) &= 2\end{aligned}\tag{1.8}$$

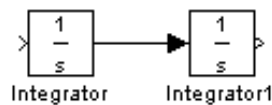
Assume that there is no input, that is, input  $u(t) = 0$ .

**PURPOSE:** *To illustrate how to configure a SIMULINK diagram for a higher order differential equation and how to introduce initial conditions into it.*

**SOLUTION:** Solve equation first with respect to the highest order derivative to obtain

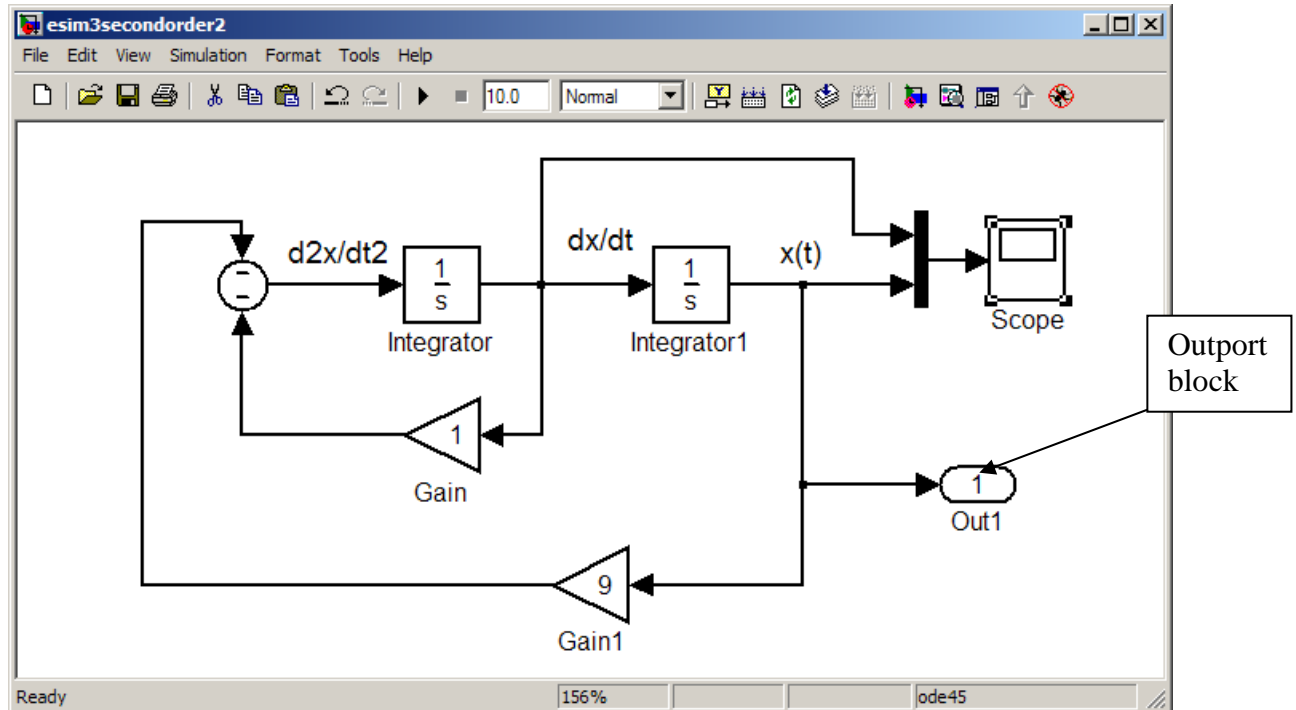
$$\frac{d^2x}{dt^2} = -5\frac{dx}{dt} - 9x\tag{1.9}$$

To set up the right-hand side two integrators are needed:



The input to the first integrator is the second derivative  $\frac{d^2x}{dt^2}$  and its output is  $\frac{dx}{dt}$ . The latter is the input to the second integrator producing  $x(t)$  at its output. In this way we have constructed the left-hand side of the equation. Since the second derivative  $\frac{d^2x}{dt^2}$  is equal to the right hand side, we collect it term by term. In order to do that we need  $\frac{dx}{dt}$  from the output of the first integrator, and  $x(t)$  from the output of the second integrator. Then  $x(t)$  must be multiplied by 9, so a *Gain* block is required. All the items are to be summed up so a *Sum* block is also needed. The final configuration is given below. The initial values are added to the integrators. The resulting configuration is given in Figure 1.25.

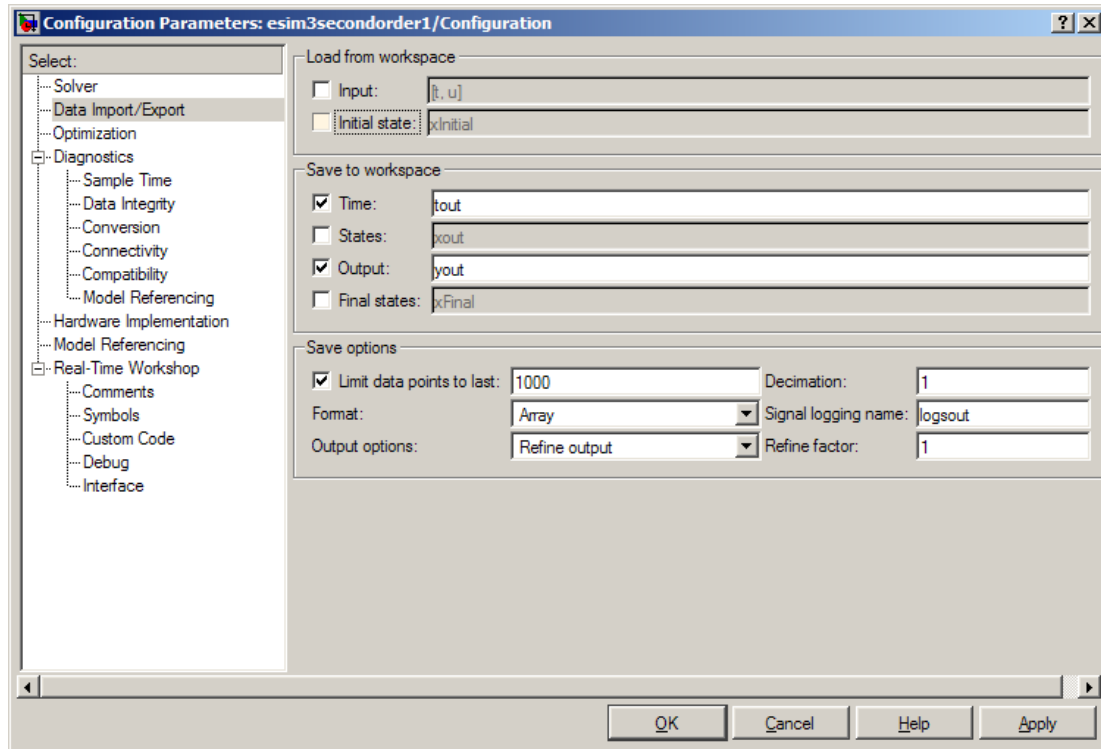




**Figure 1. 25** SIMULINK configuration of system (1.7). Multiplexer has been added so as to see  $\frac{dx}{dt}$  and  $x(t)$  at the same time. Note that one way to take information to MATLAB command window is using *Outport* block (See Figure 1.26 about details).

Finally, set up the initial conditions by clicking the integrators one at a time and make the changes according to (1.8).

To understand *Outport* block better, open *Simulation* menu and from their *Configuration parameters*. Select *Data Import/Export*, which is second on the list. By default Time *tout* and Output *yout* are saved to workspace and can be used in MATLAB command window (Figure 1.26).

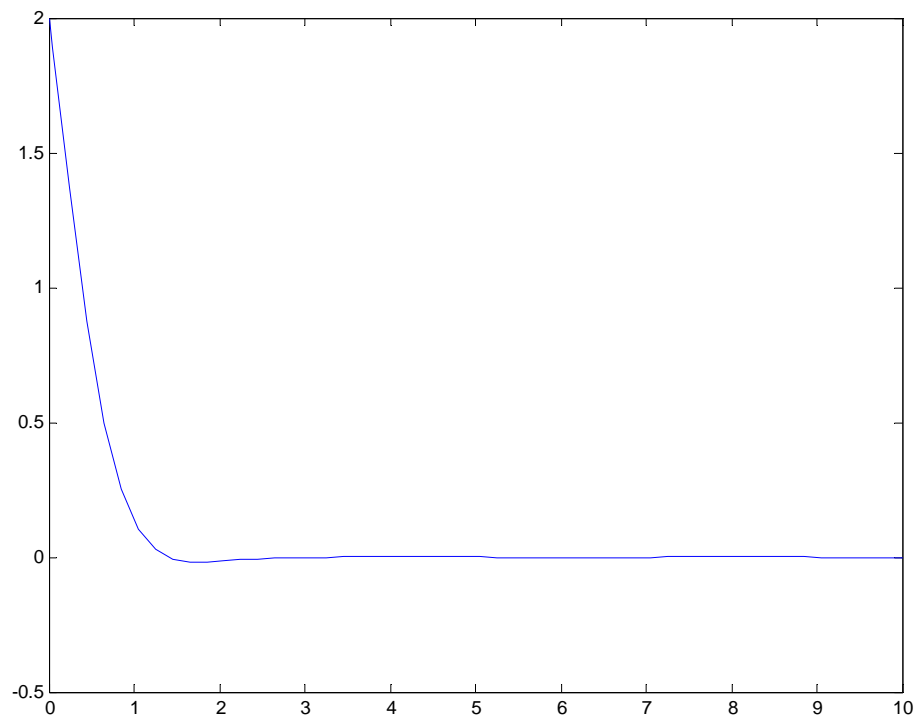


**Figure 1. 26** Configuration Parameters: Select *Data Import/Export* to see what is saved to workspace. You can also click *States*, in which case  $x(t)$  and  $\frac{dx}{dt}$  are available in the work space.

As seen from Figure 1.25, the *Outport* block corresponding to *yout*, is connected to  $x(t)$  or  $yout = x$ . For instance, plot command can now be utilized as follows

*plot(tout,yout)*

The result is displayed in Figure 1.27.

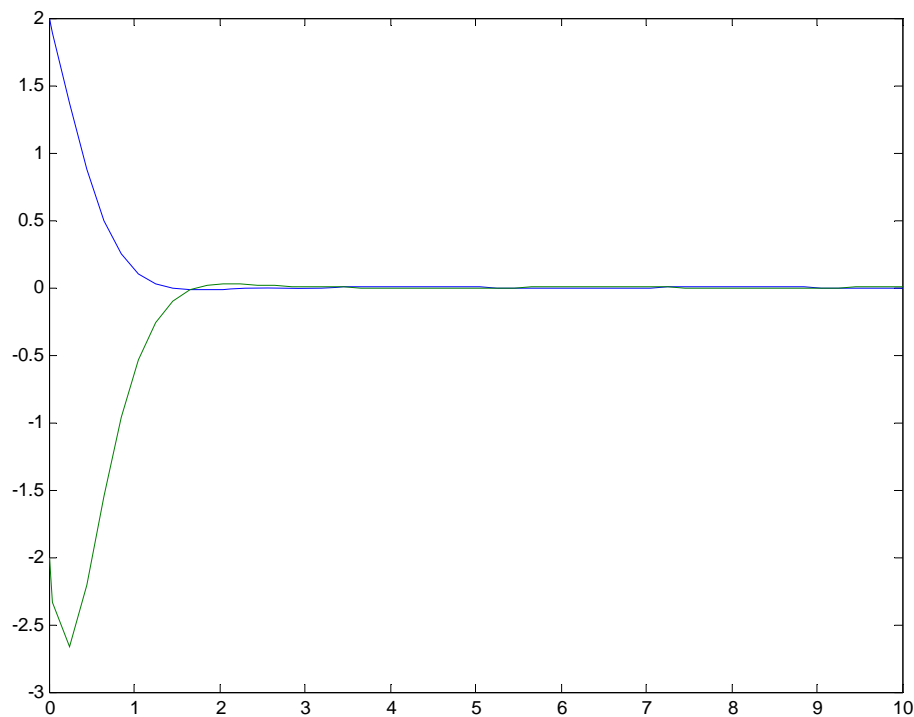


**Figure 1. 27** The simulation result  $y_{out}=x$  is plotted with *plot* command. It is possible to add labels and various other things in the figure.

If you have clicked States, then you can plot

*plot(tout,xout)*

and the result is shown in the next Figure.

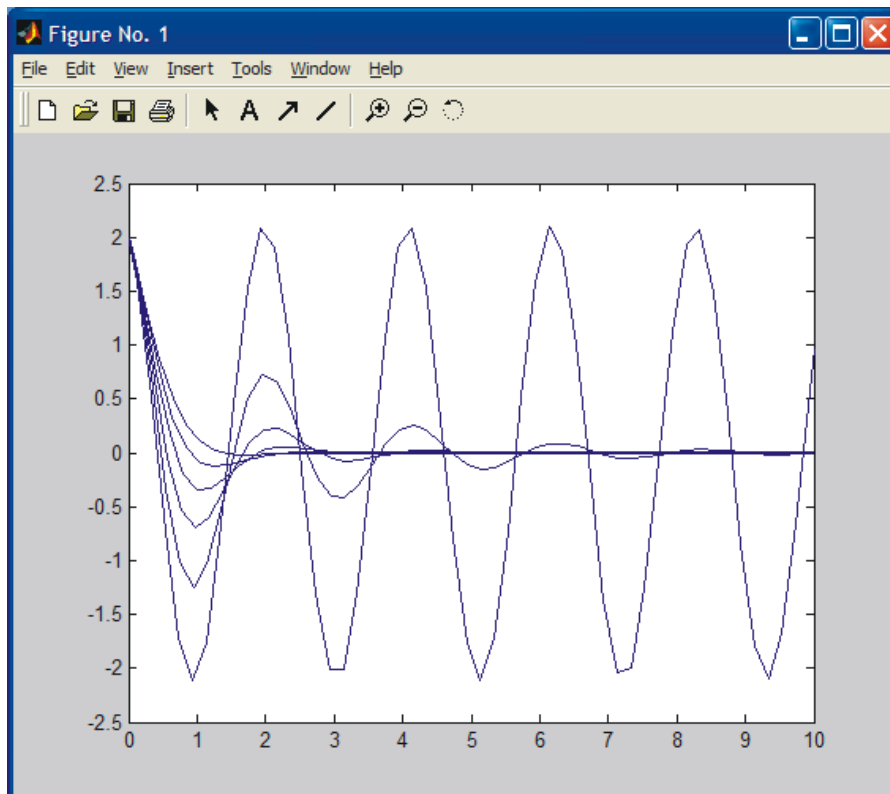


**Figure 1. 28** Simulation results of  $x(t)$  and  $\frac{dx}{dt}$  are shown,  $x(t)$  above and  $\frac{dx}{dt}$  below . This result can be produced without *Outport* by choosing *States*, but the result of Figure 1.27, that is,  $y_{out}=x$  cannot. *Instead a constant*

The same result can be seen with SIMULINK scope.

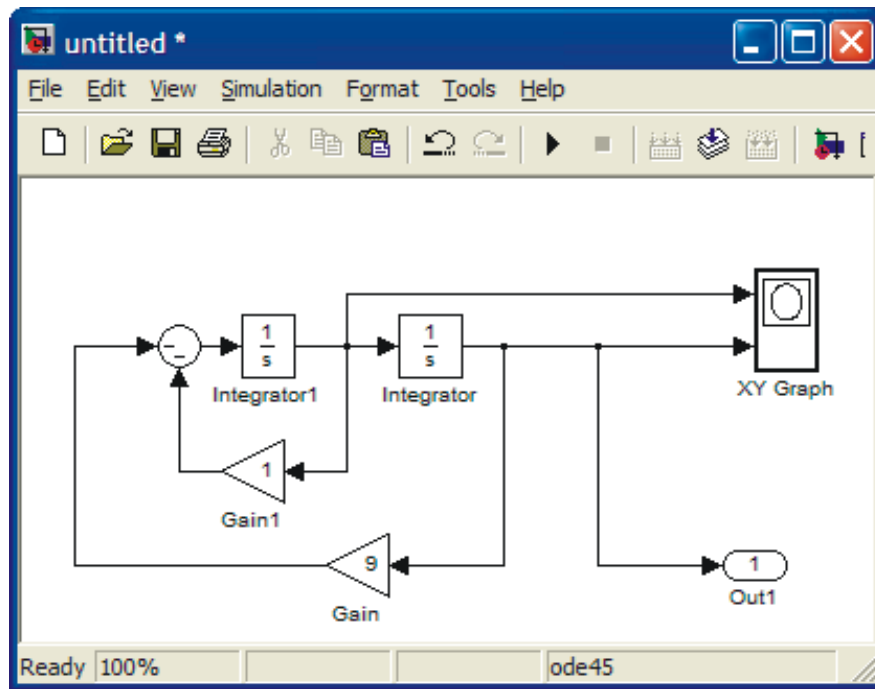
The sharpness of the lower curve around  $t = 0.4$  s is not real, it should be smooth. First you might suspect numerical difficulties (there are none) due to too large a step size. This is not the case. It is due to display graphics, i.e., not enough points have been saved to have a smooth presentation.

The damping factor can be changed by changing the coefficient 5 in front of  $\frac{dx}{dt}$ . If the coefficient is zero (no damping), the result is a sinusoidal. Increasing the damping will result in damping oscillations. Complete the study to obtain the following responses.



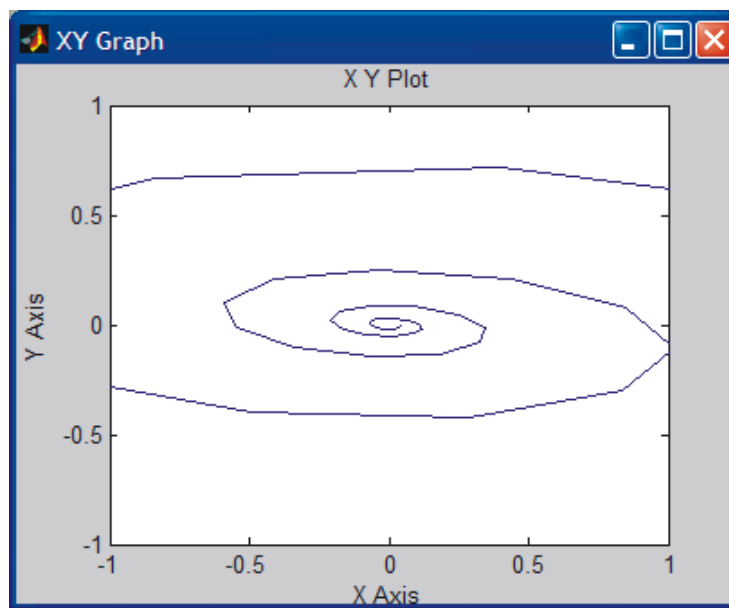
**Figure 1.29** Simulation results vary, when damping is changed. When damping constant is small, oscillation is large and when damping constant is bigger, there is no oscillation.

Let us also plot a **phase plane** plot ( $x$  vs  $dx/dt$ ). Note that the time has been eliminated and it only appears as a parameter. To see the effect better, start with less damping. Change the coefficient 5 to 1.



**Figure 1. 30** *XY Graph* scope is used to draw a phase plane plot.

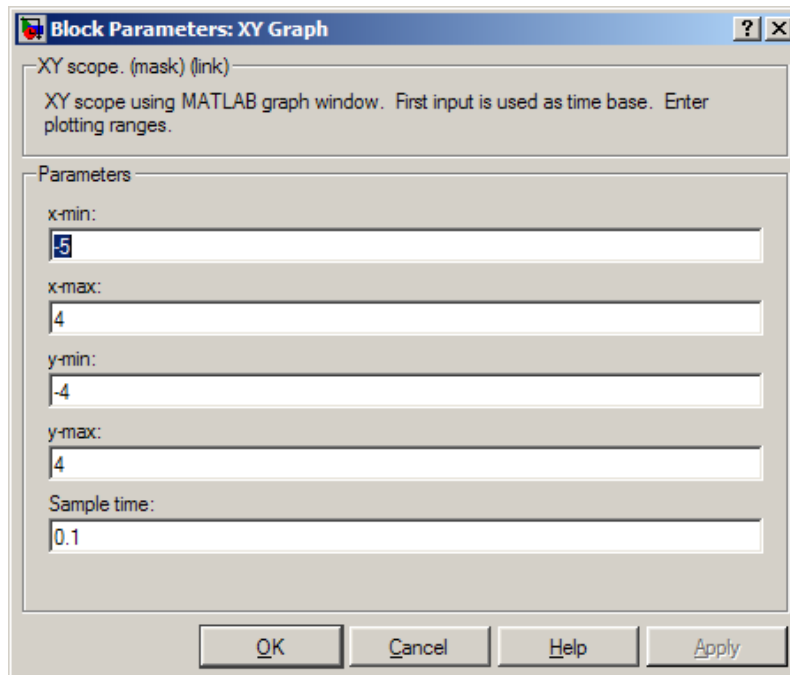
The result is shown below.



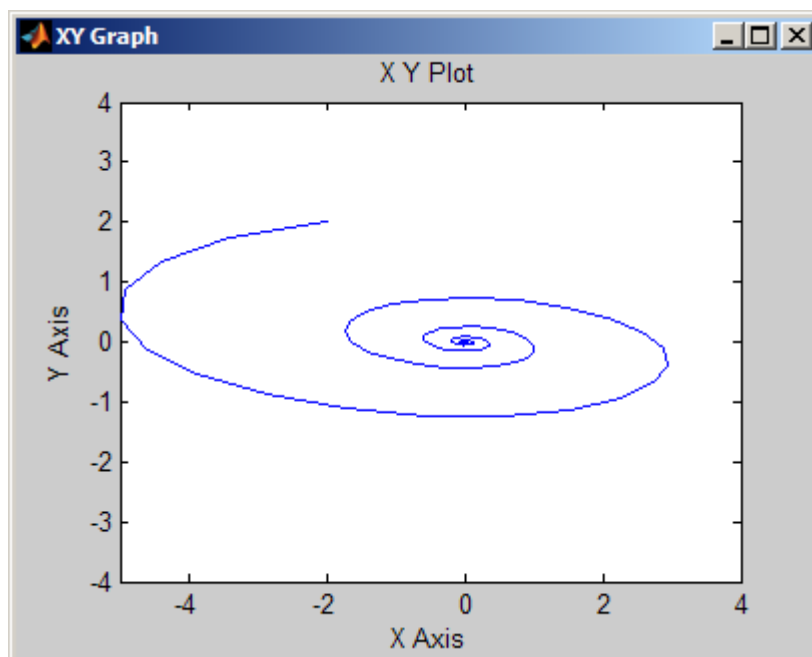
**Figure 1. 31** Phase plane plot of (1.7). The resulting figure should be smooth.

*XY Graph* does not adjust the scales automatically. In order to see the whole picture in Figure 1.31, click the *XY Graph* open and adjust the scales. Adjusting also the *Sample time* from -1 to 0.1 or

*Relative error in Configuration Parameters* from  $1e-3$  to  $1e-6$  results in a smoother Figure.



**Figure 1.32** Change the scaling *Parameters* and the sample time, in order to obtain a better figure than the one in Figure 1.31. The result is shown in Figure 1.33.



**Figure 1.33** The whole, smooth phase plane plot is displayed. A stable system. It converges towards the origin. Physical interpretation: In the origin both position and velocity are zero.