

Short Tutorial on Matlab

(©2004, 2005 by Tomas Co)

Part 3. Simulink® Basics

1. A simple example.

Suppose you want to model the response of a first order process model given by the following equation:

$$\frac{dT}{dt} = \frac{1}{\tau}(T_{in} - T)$$
$$T(0) = T_o$$

where τ is the residence time parameter, T_{in} is the inlet temperature and T is the temperature of the continuously stirred tank.

Step 1. Activate SIMULINK

In the command line, type

```
>> simulink
```

(Alternatively, you can use the Matlab launch pad and double click on Simulink icon.)

The simulink library browser should pop out as shown in Figure 1.

Step 2. Create a blank Simulink model window.

On the Library Browser window double-click on the "Create Model" menu button (as indicated in Figure 1.). Alternatively, you could also select from the **[File]→[New]→[Model]** submenu choice.

A model window as shown in Figure 2 should now pop out. This is where we will be adding our simulation blocks.

Step 3. Import blocks from the Library Browser to the Model window.

Going back to Figure 1, double-click on the "**Math Operations**" icon. The right-side of the Library browser should change. Scroll down until you find the "**Gain**" block (as shown in Figure 3.)

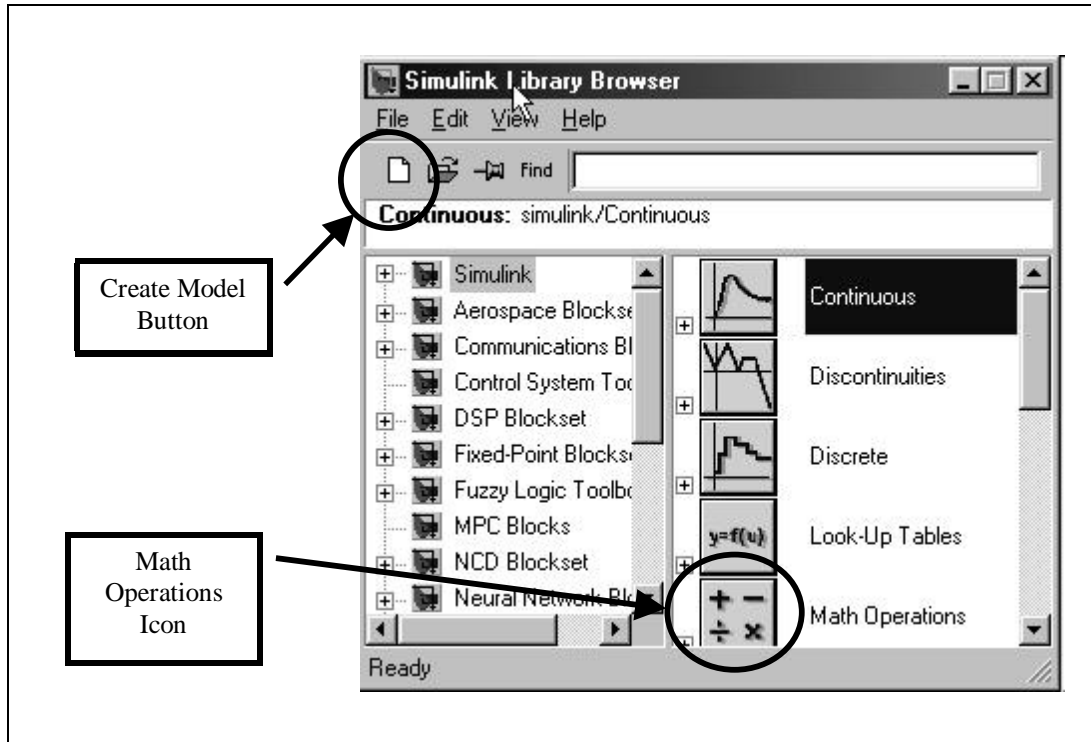


Figure 1.

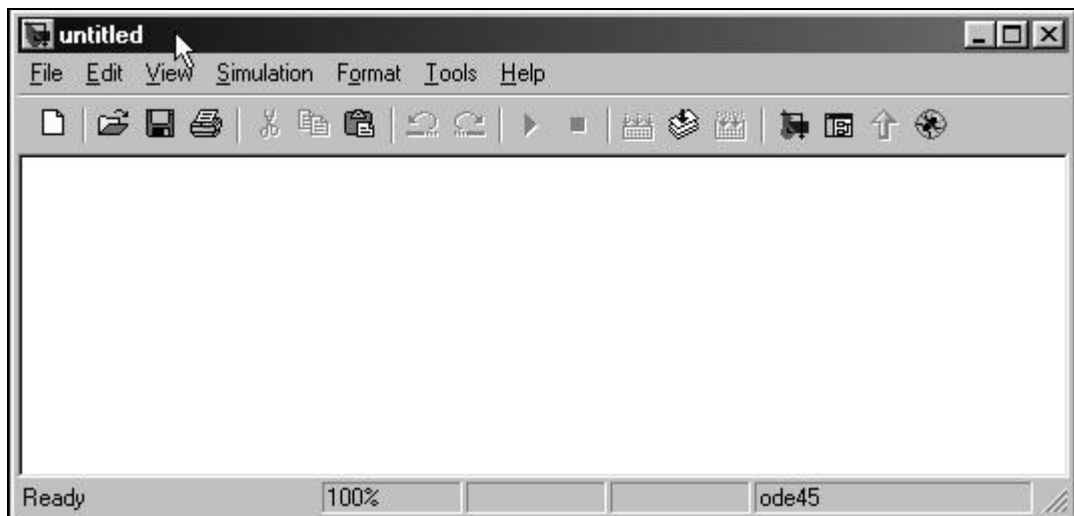


Figure 2.

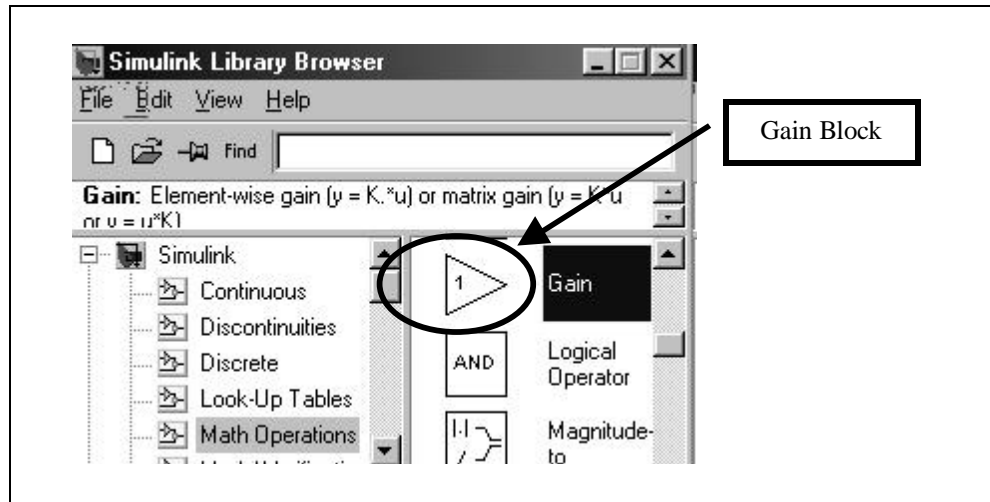


Figure 3.

Now drag-drop the "**Gain**" block into the Model window.

Back to the Library browser, on the right side, scroll down further until you find the "**Sum**" block and then drag-drop this block into the Model window.

You could move the two blocks (via click-drag) around in the model window to match what is shown in Figure 4.

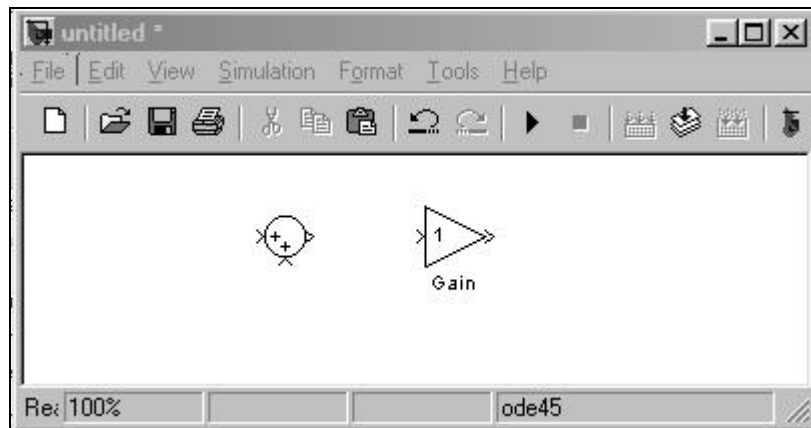


Figure 4.

We need three more blocks which are included in other Simulink subdirectories. First, click on the "**Continuous**" subdirectory that is on the right side of the Library Browser. This should change the left side of the Library Browser window as shown in Figure 5. Select the "**Integrator**" block and drag-drop it into the Model window.

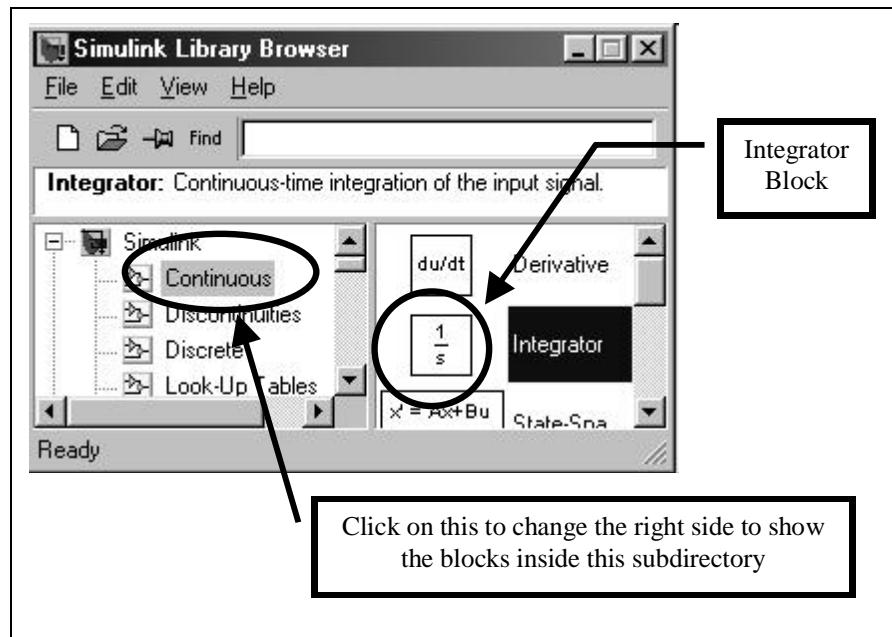


Figure 5.

Next, in the Library Browser, select the "**Sources**" subdirectory (left side). On the right side of the Browser, select the "**Step**" block and drag-drop it into the Model window.

Finally, in the Library Browser, select the "**Sinks**" subdirectory (left side). On the right side of the Browser, select the "**Scope**" block and drag-drop it into the Model window.

After all these blocks have been imported, the blocks can be moved around to match the positions shown in Figure 6.

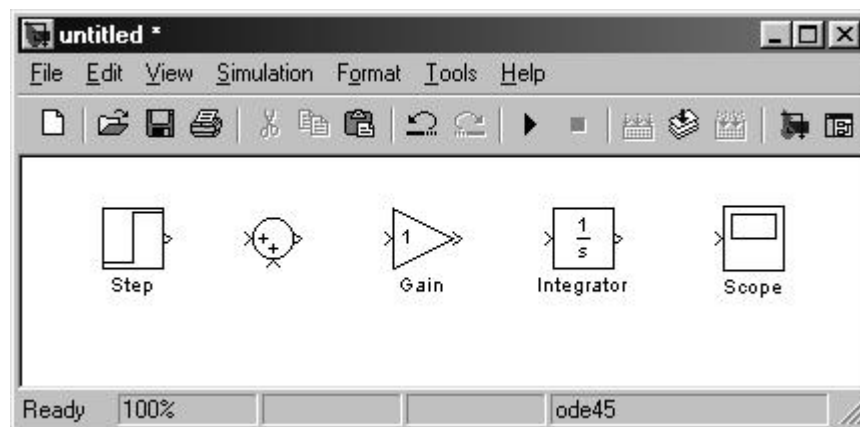


Figure 6.

At this point, it may be instructive to layout the roles and connections among these five blocks:

- i) The **Step** block will be used to implement a step change in T_{in} .
- ii) The **Sum** block will be used to take the difference: $(T_{in}-T)$. Of course, we need to change the properties of this block to reflect a difference instead of a sum (which we will do below).
- iii) The **Gain** block will be used to change the difference $(T_{in}-T)$ by a factor. In our case, this factor will be $(1/\tau)$. Thus the output of the **Gain** block will be $(1/\tau)(T_{in}-T)$. In reference to our process model, the calculations yield the value of the derivative dT/dt .
- iv) The **Integrator** block now takes in the dT/dt signal and outputs the desired T signal.
- v) The **Scope** block now simply reads in the T signals as a function of time and plots it in a separate window.

Step 4. Change the properties of the blocks, if needed.

As we mentioned, we need to change the **Sum** block to obtain a difference. To do so, double-click the **Sum** block. A window should pop-up as shown in Figure 7. In the "List of Signs" input box, change the original entry to "**|+-**". Then click the [OK] button to close this window and go back to the Model window.

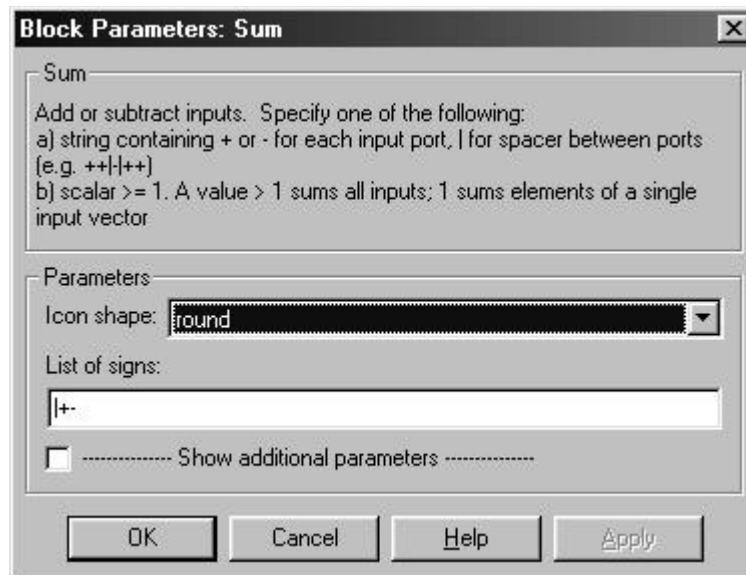


Figure 7.

Next, double-click the **Step** block. A parameter window should pop-up. Match the parameters shown in Figure 8. This means T_{in} will have a value of 20 until the time, $t=3$. After that, T_{in} will jump to a value of 30.

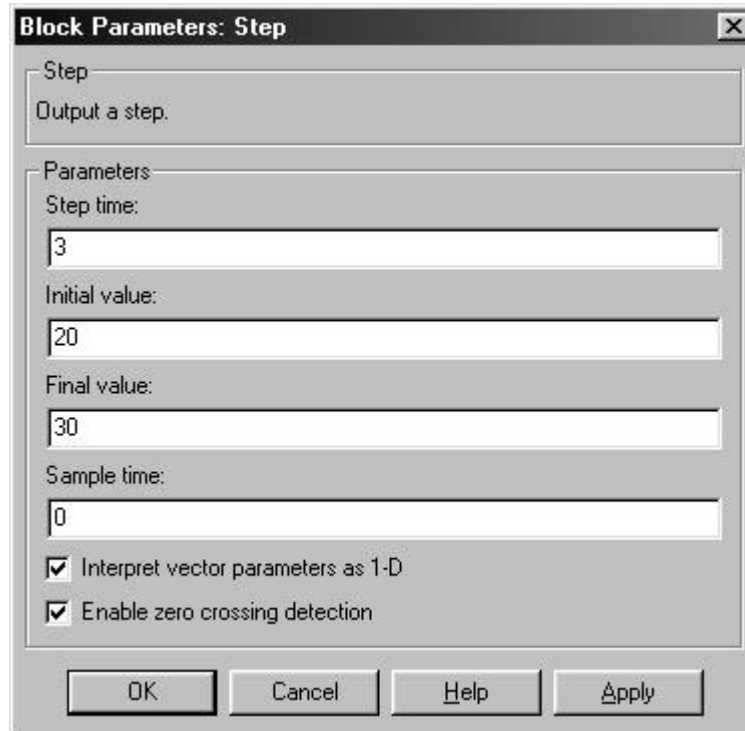


Figure 8.

Next, double-click the **Integrator** block. A parameter window should pop-up. In the "Initial value:" input box, change it to 20. This means we are setting $T_o=20$.

Finally, double-click the **Gain** Block. Another parameter window should pop-out. Suppose we want to use a value of $\tau = 4$ for our time constant. This means the reciprocal, $(1/\tau)=0.25$. Thus, in the parameter window for the **Gain** block, change the gain from 1 to 0.25, and click [OK]. (In the Model window, the value of 0.25 might not show. You can resize the **Gain** block by clicking on it once, and then dragging the black corners to resize it.)

Step 5. Connect the Blocks.

Method 1: Short cut.

To connect the **Integrator** block and the **Scope** block, first click on the Integrator block. Next, CTRL-click on the **Scope** block. An arrow should connect the two blocks. (Note: the sequence of blocks clicked will determine the direction of the arrow.)

Method 2: Manual.

Try connecting the **Gain** block with the **Integrator** block. First, move the cursor to the left port of the **Gain** block until the cursor changes to a cross symbol. Next, click-drag the cursor to the input port of the **Integrator** (a dotted line should be dragging behind), then release the mouse-click.

Using either method 1 or method 2 to connect the blocks to match Figure 9.

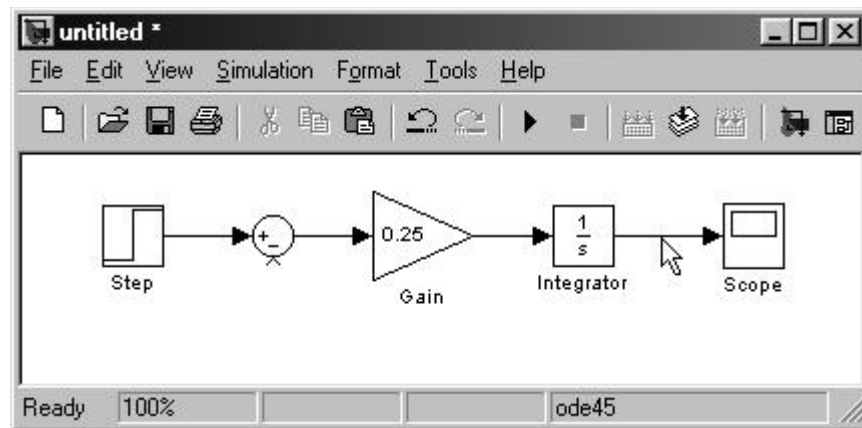


Figure 9.

Next, we need to tap into (i.e. split) the T signal that goes to the **Scope** block, and feed it back to the **Sum** block. First, position the cursor somewhere in the middle portion of the signal line connecting the Integrator block and the **Scope** block as shown in Figure 9. Next, while depressing the CTRL key, click-drag the cursor (which should turn to a cross once you depressed the right button of the mouse) until it is positioned on top of the "minus" port of the **Sum** block, then release the buttons. A new split line should appear. (You can resize the signal line by clicking on it once and then dragging the black corners.) We now have the final configuration for our model shown in Figure 10.

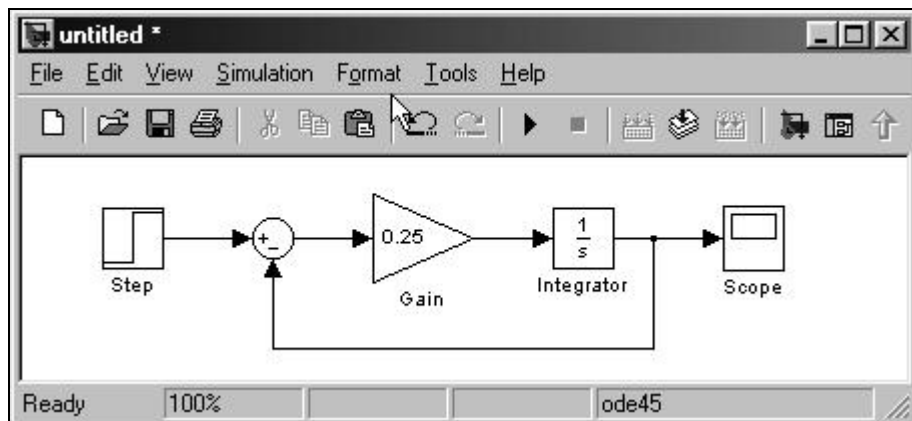


Figure 10.

Step 5. Perform the simulation.

We could set simulation parameters by first selecting the submenu item [Simulation]→[Simulation parameters...]

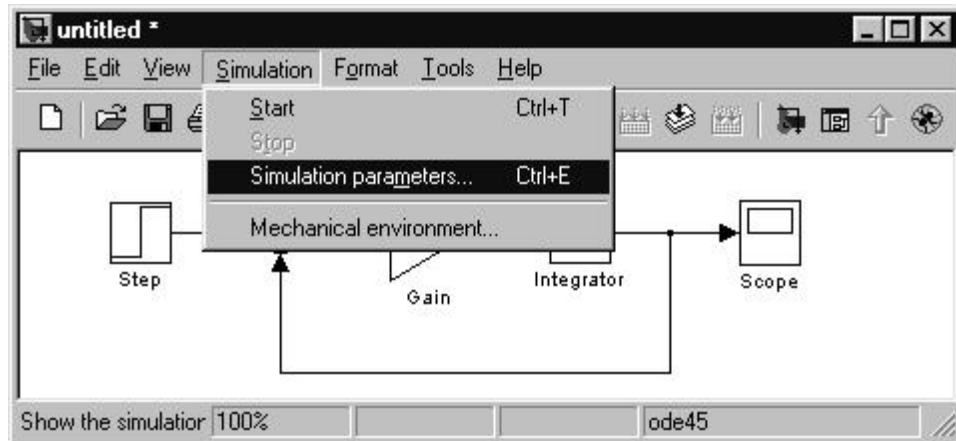


Figure 11.

(for version 7.0, this will be [Configuration parameters...])

Set the stop time to 20.0 as shown in Figure 12, and then click [OK].

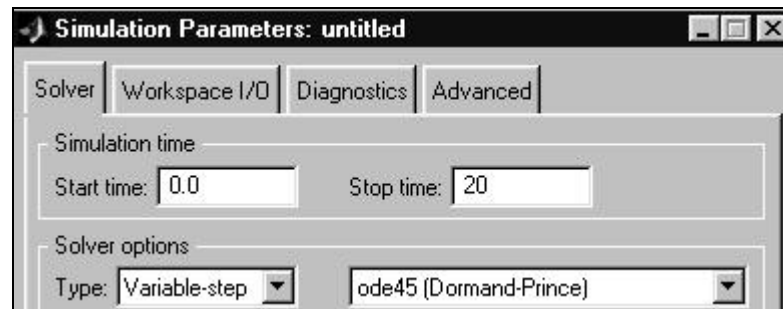


Figure 12.

Next, run the simulation by pressing the “Run” button as shown in Figure 13. Alternatively, you can select the [Simulation]→[Start] submenu item (see Figure 11.)

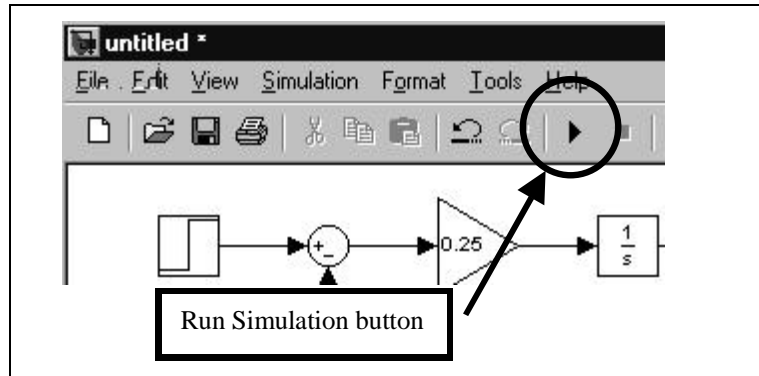


Figure 13.

To see the results, double-click the **Scope** block. A figure with a plot should pop out. To see the whole plot, click the “Autoscale” button, as shown in Figure 14.

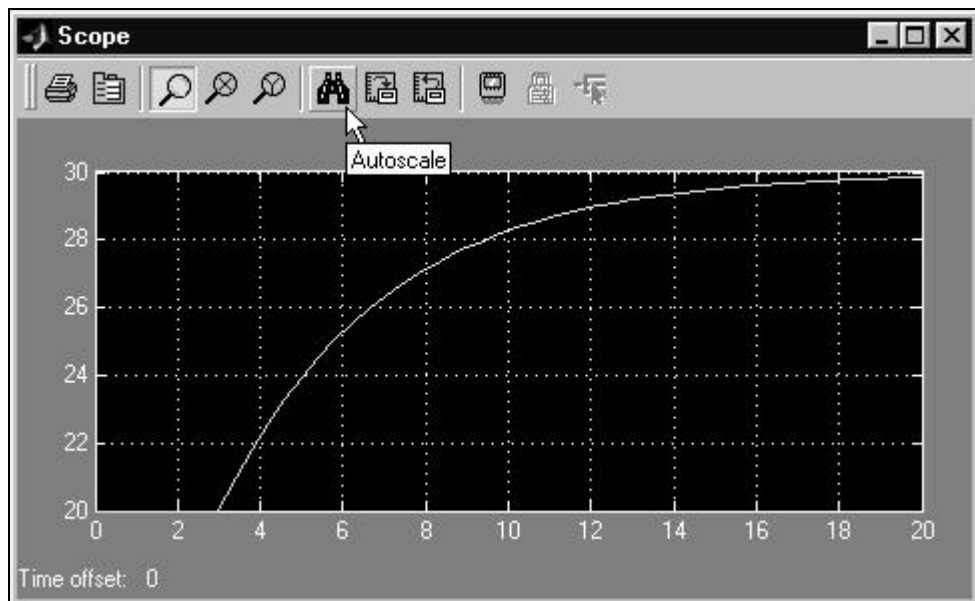


Figure 14.

Step 6. Save the model.

In the Model window, select [File]→[Save As] and save the model. The file will be save with a *.mdl extension. (For later purposes, we will save our model system as **simple.mdl**.)

2. Communication between the Matlab's Command window/workspace and Simulink's Model window.

a) Exporting the outputs to the workspace.

Go to the Simulink Library Browser and select the **sinks** subdirectory (on the left side). From the right side of the browser, drag-drop the [To Workspace] block into the Model window and drag another split signal to this block as shown in Figure 14.

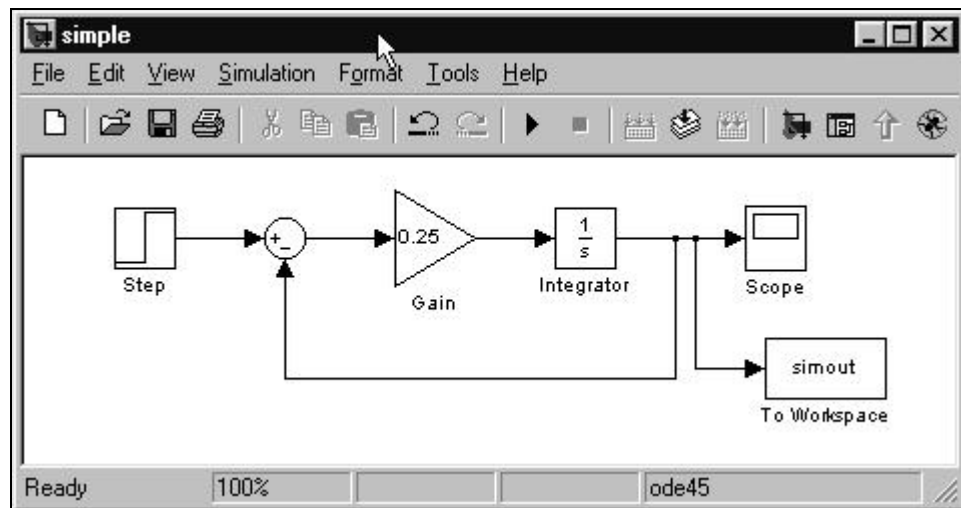


Figure 14.

Double-click the [To Workspace] block (in Figure 14). A parameter window should pop-out. Change the entry in **Variable Name** to T. Also, change the **Save Format** selection to “**Array**”, as shown in Figure 15. Click [OK].

Next, in the Model window, select the [Simulation]→[simulation parameters...] submenu once more. This time choose the [Workspace I/O] tab. Click on the **Time** variable and change the name from **tout** to **time** as shown in Figure 16. Also, near the bottom of the window, change the **Format** to: **Array**. Click [OK]. (In version 7.0, you need to select [Data import/export])

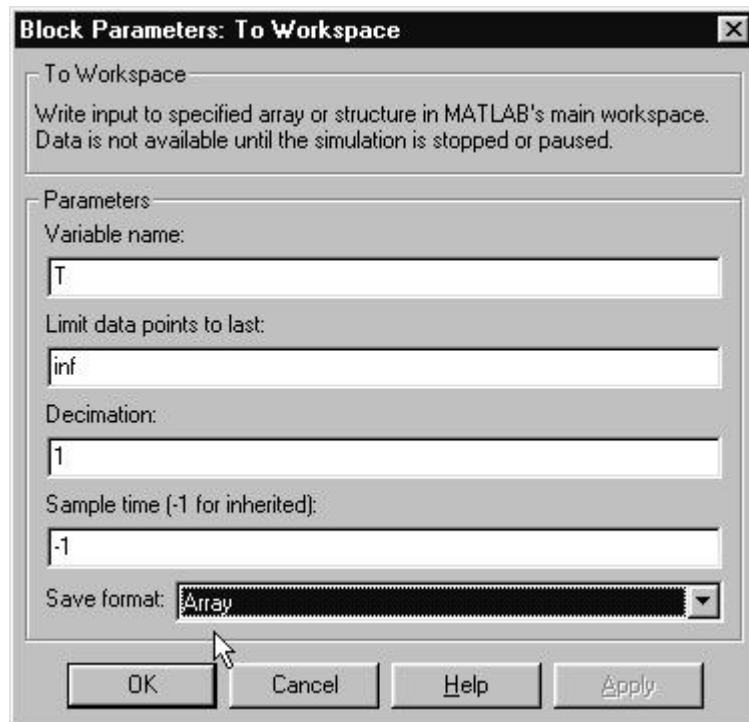


Figure 15.

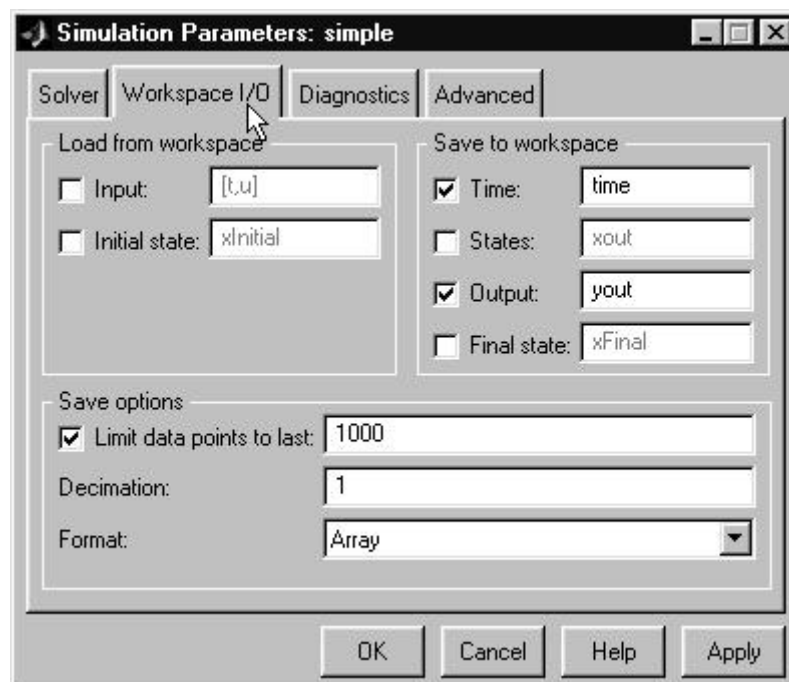


Figure 16.

Start the simulation by clicking the “**Run**” button. Next, go back to the Matlab command window. You should now have the two vectors available in the workspace: **time** and **T**, which you can plot or perform other forms of data processing via Matlab commands.

b) Invoking a Simulink run from Matlab command window.

Sometimes, it may be more efficient to run the simulation from the command window. For instance, you could change parameters for a range of say 10 values using a script file and then collect the results in a structure or a file for further processing.

- To read the value of 'Gain' parameter in the **Gain** block, use the following command:

```
>> g_svalue=get_param('simple/Gain','Gain')
```

Remarks:

- i) The string, '**simple/Gain**', identifies the **Gain** block in the model we called simple (since we saved the model as **simple.mdl** earlier).

Tip: If you are unsure about the path name for a block, you can first go to the model window and click on the block of interest. Then go back to the Matlab command window and type the command **gcb**, (which stands for "get current block"):

```
>> gcb  
  
ans =  
  
simple/Gain
```

- ii) The second string, '**Gain**', is one of the parameters in the parameter window. Another parameter you could try to read is '**Multiplication**', among others.

Note: This command line will yield a string result. To change the string to a numeric value, you will need to use the Matlab function: **str2num**, i.e.

```
>> g_value=str2num(g_svalue)  
  
g_value =  
  
0.2500
```

- To change the parameter value in a Simulink Model while in a Matlab command window, use `set_param` :

```
>> set_param('simple/Gain','Gain','0.30')
```

Note: The input for the 'Gain' parameter is a string, hence we used '0.30'.

If you go back to the model window, the value shown in the **Gain** block should have changed from 0.25 to 0.30.

- To run the Simulink simulation from a command window, type the following:

```
>> sim('simple')
```

- **Example:** Suppose you want to run the Simulink model, **simple**, which we have built so far, for several values of gain and graph the results all together in a single plot. One way is to use create a script file as show below and run it.

```

%
%      Script file for running the Simulink model 'simple'
%      for different values of gain

%      (c) 2004 Tom Co
%      @ Michigan Technological Univesity

open_system('simple')

% Initializations
% =====
tagCollect =[] ;
timeCollect =[] ;
TCollect =[] ;
plotcolors ='bgcmkbgcmkbgcmk' ;
nvals = 0 ;

% Calculations
% =====
for gval=0.2:0.2:1.0
    nvals = nvals + 1 ;
    gvalStr = num2str(gval,3) ;
    set_param('simple/Gain','Gain',gvalStr) ;
    sim('simple') ;
    timeCollect = [timeCollect, time] ;
    TCollect = [TCollect, T] ;
    tag = ['Gain = ',gvalStr] ;
    tagCollect = [tagCollect,{tag}] ;
end

% Plotting
% =====
hold off
for i=1:nvals
    plot(timeCollect(:,i),TCollect(:,i),plotcolors(i));
    hold on
end
hold off

legend(tagCollect) ;
xlabel('Time (mins)') ;
ylabel('Temperature (^oF)') ;

```

Remarks:

- i) We included a line :

```
open_system('simple')
```

This is to make sure the Model is open prior to the use of commands **set_param** and **sim**. (If the model, **simple**, is already open, it will not do anything.)

- ii) The variable **plotcolors** is a string array that would indicate the colors used for plotting. For example, **'bcg'** will mean the sequence black, cyan, green.
- iii) **tagCollect** is a cell array, so be careful to use the curly brackets as written in the script. We use a cell array because this is what is required by the command **legend**.
- iv) Note the use of commands, **hold on** and **hold off** to allow the plot command not to erase previous plots.
- v) Since when using **set_param**, the gain value should be a string, we need to convert the numeric value of **gval** to a string as follows:

```
gvalStr = num2str(gval,3) ;
```

where 3 signifies the string will show three significant figures.