



## Getting started with Simulink

The following tutorial gives a quick introduction to Simulink for those that have not worked with Simulink before. It takes about 30 min to complete.

## What Is Simulink?

Simulink is a software package for modeling, simulating, and analyzing dynamic systems. It supports linear and nonlinear systems, modeled in continuous time, sampled time, or a hybrid of the two. Systems can also be multirate, i.e., have different parts that are sampled or updated at different rates.

### Tool for Simulation

Simulink encourages you to try things out. You can easily build models from scratch, or take an existing model and add to it. You have instant access to all the analysis tools in MATLAB®, so you can take the results and analyze and visualize them. A goal of Simulink is to give you a sense of the *fun* of modeling and simulation, through an environment that encourages you to pose a question, model it, and see what happens.

Simulink is also practical. With thousands of engineers around the world using it to model and solve real problems, knowledge of this tool will serve you well throughout your professional career.

### Tool for Model-Based Design

With Simulink, you can move beyond idealized linear models to explore more realistic nonlinear models, factoring in friction, air resistance, gear slippage, hard stops, and the other things that describe real-world phenomena. Simulink turns your computer into a lab for modeling and analyzing systems that simply wouldn't be possible or practical otherwise, whether the behavior of an automotive clutch system, the flutter of an airplane wing, the dynamics of a predator-prey model, or the effect of the monetary supply on the economy. Simulink provides numerous demos that model a wide variety of such real-world phenomena. For more information about accessing and executing these demos, see Chapter 2, “Running a Model”.

For modeling, Simulink provides a graphical user interface (GUI) for building models as block diagrams, using click-and-drag mouse operations. With this interface, you can draw the models just as you would with pencil and paper (or as most textbooks depict them). This is a far cry from previous simulation packages that require you to formulate differential equations and difference equations in a language or program. Simulink includes

a comprehensive block library of sinks, sources, linear and nonlinear components, and connectors. You can also customize and create your own blocks. For information on creating your own blocks, see [Writing S-Functions](#) in the online documentation.

Models are hierarchical, so you can build models using both top-down and bottom-up approaches. You can view the system at a high level, then double-click blocks to go down through the levels to see increasing levels of model detail. This approach provides insight into how a model is organized and how its parts interact.

After you define a model, you can simulate it, using a choice of integration methods, either from the Simulink menus or by entering commands in the MATLAB Command Window. The menus are particularly convenient for interactive work, while the command-line approach is very useful for running a batch of simulations (for example, if you are doing Monte Carlo simulations or want to sweep a parameter across a range of values). Using scopes and other display blocks, you can see the simulation results while the simulation is running. In addition, you can change many parameters and see what happens for “what if” exploration. The simulation results can be put in the MATLAB workspace for postprocessing and visualization.

Model analysis tools include linearization and trimming tools, which can be accessed from the MATLAB command line, plus the many tools in MATLAB and its application toolboxes. And because MATLAB and Simulink are integrated, you can simulate, analyze, and revise your models in either environment at any point.

## **Related Products**

The MathWorks provides several products that are especially relevant to the kinds of tasks you can perform with Simulink and that extend the capabilities of Simulink. For information about these related products, see <http://www.mathworks.com/products/simulink/related.html>.

# Running a Model

---

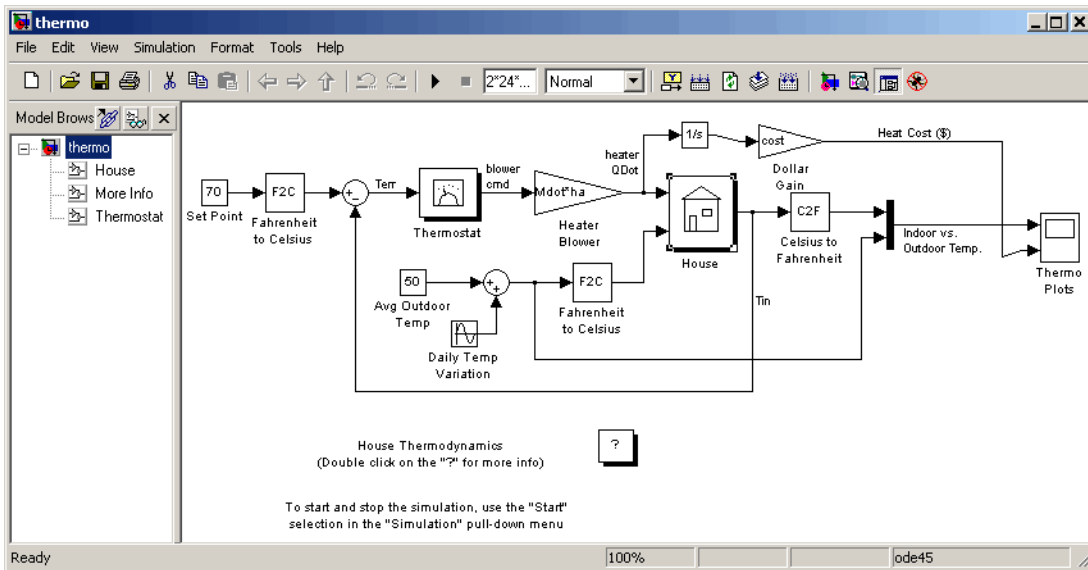
The following section gives you a quick introduction to running a Simulink® model.

Running a Demo Model (p. 2-2)	Example of how to run a Simulink model.
About the Demo Model (p. 2-4)	Explains how the model works.
Some Things to Try (p. 2-7)	Suggests some model parameters to change to see how they affect the simulation.
What This Demo Illustrates (p. 2-7)	Describes the common modeling and simulation tasks that this demo illustrates.
Other Useful Demos (p. 2-8)	Shows how to find other demos that illustrate key Simulink concepts and features.

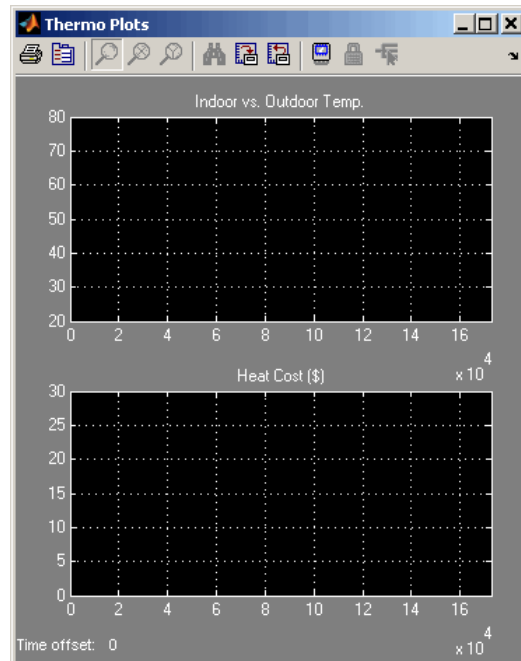
## Running a Demo Model

An interesting demo program provided with Simulink models the thermodynamics of a house. To run this demo, follow these steps:

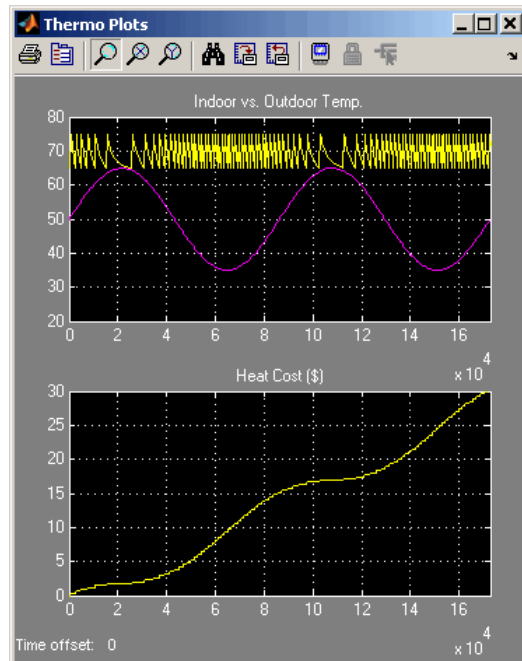
- 1 Start MATLAB. See your MATLAB documentation if you're not sure how to do this.
- 2 Run the demo model by typing thermo in the MATLAB Command Window. This command starts up Simulink and creates a model window that contains this model.



It also creates a scope window for the model. The Scope block displays two plots labeled Indoor vs. Outdoor Temp and Heat Cost (\$), respectively.



- 3 To start the simulation, pull down the **Simulation** menu and choose the **Start** command (or, on Microsoft Windows, click the **Start** button on the Simulink toolbar). As the simulation runs, the indoor and outdoor temperatures appear in the Indoor vs. Outdoor Temp plot as yellow and magenta signals, respectively. The cumulative heating cost appears in the Heat Cost (\$) plot.



See Scope for further details about signal element colors.

- 4 To stop the simulation, choose the **Stop** command from the **Simulation** menu (or click the **Pause** button on the toolbar). If you want to explore other parts of the model, look over the suggestions in “Some Things to Try” on page 2-7.
- 5 When you’re finished running the simulation, close the model by choosing **Close** from the **File** menu.

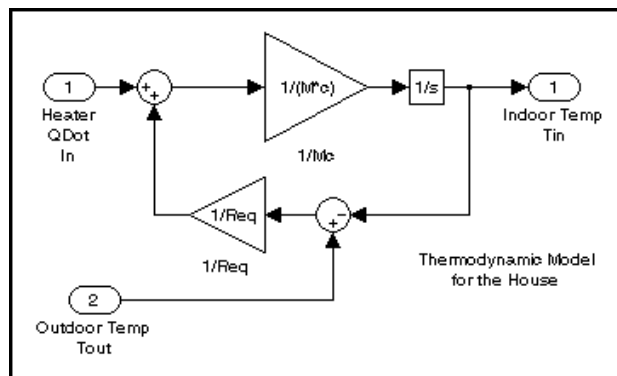
## About the Demo Model

The demo models the thermodynamics of a house. The thermostat is set to 70 degrees Fahrenheit and is affected by the outside temperature, which varies by applying a sine wave with amplitude of 15 degrees to a base temperature of 50 degrees. This simulates daily temperature fluctuations.



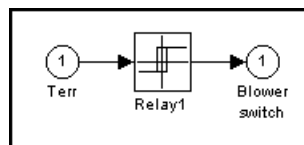
The model uses subsystems to simplify the model diagram and create reusable systems. A subsystem is a group of blocks that is represented by a Subsystem block. This model contains five subsystems: one named Thermostat, one named House, and three Temp Convert subsystems (two convert Fahrenheit to Celsius, one converts Celsius to Fahrenheit).

The internal and external temperatures are fed into the House subsystem, which updates the internal temperature. Double-click the House block to see the underlying blocks in that subsystem.



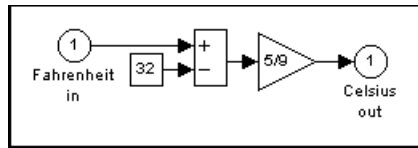
House subsystem

The Thermostat subsystem models the operation of a thermostat, determining when the heating system is turned on and off. Double-click the block to see the underlying blocks in that subsystem.



Thermostat subsystem

Both the outside and inside temperatures are converted from Fahrenheit to Celsius by identical subsystems. Note that the conversion subsystems are masks. Right-click the subsystem and select **Look Under Mask** to view the subsystem (see "Creating Masked Subsystems" for further details).



**Fahrenheit to Celsius conversion (F2C)**

When the heat is on, the heating costs are computed and displayed on the Heat Cost (\$) plot on the Thermo Plots Scope. The internal temperature is displayed on the Indoor Temp Scope.

## Some Things to Try

Here are several things to try to see how the model responds to different parameters:

- Each Scope block contains one or more signal display areas and controls that enable you to select the range of the signal displayed, zoom in on a portion of the signal, and perform other useful tasks. The horizontal axis represents time and the vertical axis represents the signal value.
- The Constant block labeled Set Point (at the top left of the model) sets the desired internal temperature. Open this block and reset the value to 80 degrees. Rerun the simulation to see how the indoor temperature and heating costs change. Also, adjust the outside temperature (the Avg Outdoor Temp block) and rerun the simulation to see how it affects the indoor temperature.
- Adjust the daily temperature variation by opening the Sine Wave block labeled Daily Temp Variation and changing the **Amplitude** parameter and rerun the simulation.

## What This Demo Illustrates

This demo illustrates several tasks commonly used when you are building models:

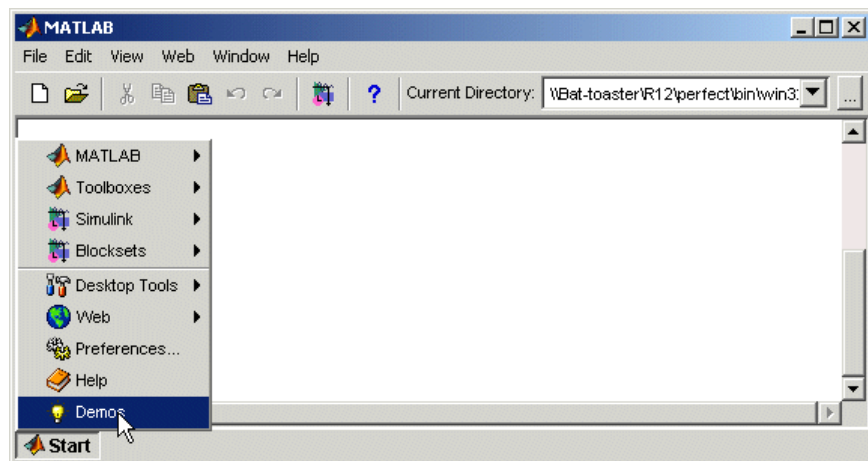
- Running the simulation involves specifying parameters and starting the simulation with the **Start** command, described in “Running Simulations”.
- You can encapsulate complex groups of related blocks in a single block, called a subsystem. See “Creating Subsystems” for more information.
- You can customize the appearance of and design a dialog box for a block by using the masking feature, described in detail in “Creating Masked Subsystems”. The thermo model uses the masking feature to customize the appearance of all the Subsystem blocks that it contains.
- The Scope blocks display graphic output much as an actual oscilloscope does.

### Other Useful Demos

Other demos illustrate useful modeling concepts. You can access these demos from the MATLAB Command Window:

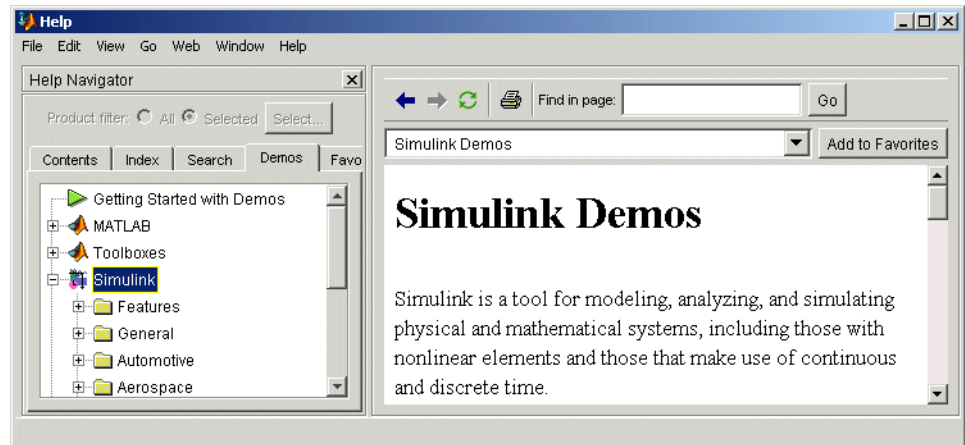
- 1 Click the **Start** button on the bottom left corner of the MATLAB Command Window.

The **Start** menu appears.



- 2 Select **Demos** from the menu.

The MATLAB Help browser appears with the **Demos** pane selected.



**3** Click the **Simulink** entry in the **Demos** pane.

The entry expands to show groups of Simulink demos. Use the browser to navigate to demos of interest. The browser displays explanations of each demo and includes a link to the demo itself. Click on a demo link to start the demo.



# Building a Model

---

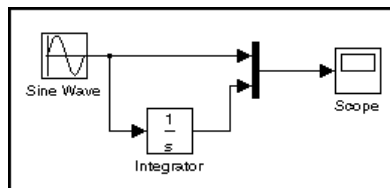
The following sections show you to build a model of a simple dynamic system, using Simulink.

Model Example (p. 3-2)	Example of how to build a Simulink model.
Creating an Empty Model (p. 3-3)	How to create an empty model.
Adding Blocks (p. 3-5)	How to add blocks to a model.
Connecting the Blocks (p. 3-9)	How to connect block outputs to block inputs.
Configuring the Model (p. 3-12)	How to select model configuration options.
Running the Model (p. 3-13)	How to run a Simulink model.

## Model Example

This example shows you how to build a model using many of the model-building commands and actions you will use to build your own models. The instructions for building this model in this section are brief. All the tasks are described in more detail in the next chapter.

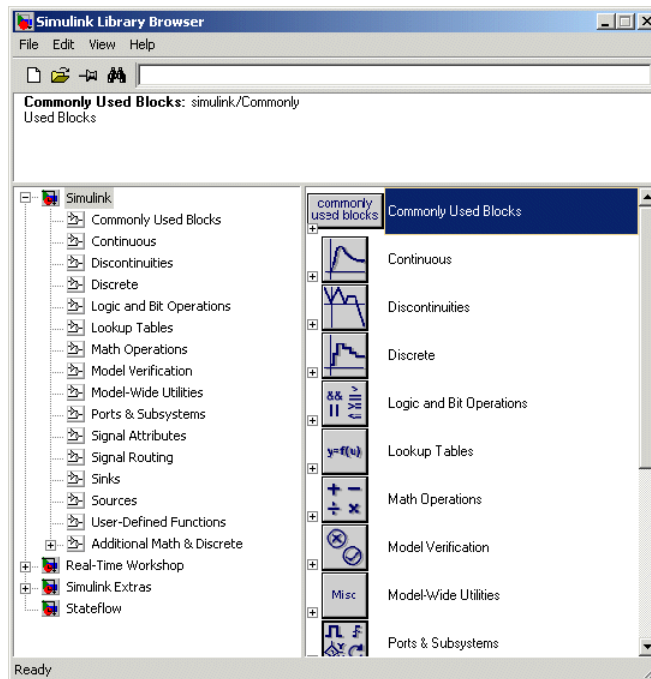
The model integrates a sine wave and displays the result along with the sine wave. The block diagram of the model looks like this.





## Creating an Empty Model

To create the model, first enter `simulink` in the MATLAB Command Window. On Microsoft Windows, the Simulink Library Browser appears.



## Adding Blocks

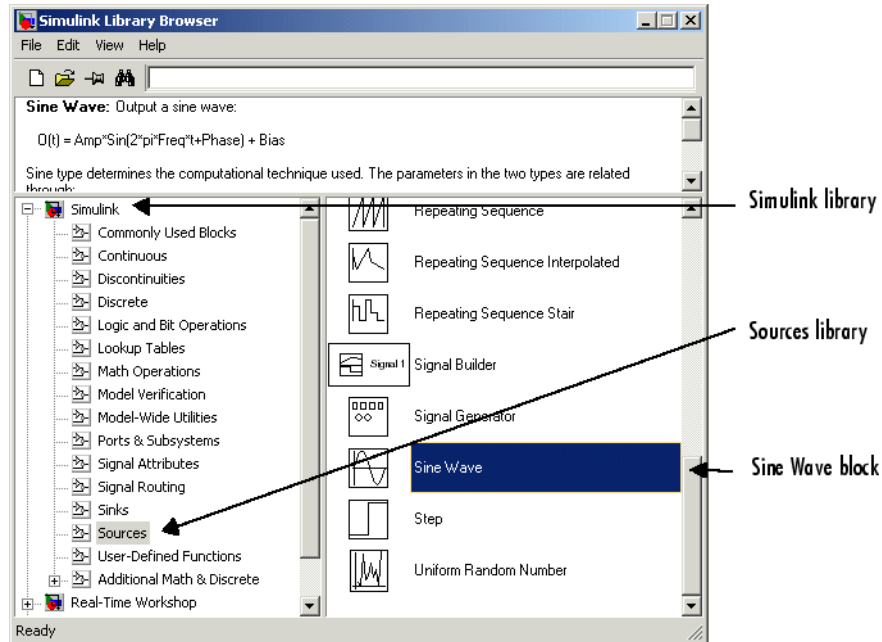
To create this model, you need to copy blocks into the model from the following Simulink block libraries:

- Sources library (the Sine Wave block)
- Sinks library (the Scope block)
- Continuous library (the Integrator block)
- Signal Routing library (the Mux block)

You can copy a Sine Wave block from the Sources library, using the Library Browser (Windows only) or the Sources library window (UNIX, and Windows).

To copy the Sine Wave block from the Library Browser, first expand the Library Browser tree to display the blocks in the Sources library. Do this by clicking the Sources node to display the Sources library blocks. Finally, click the Sine Wave node to select the Sine Wave block.

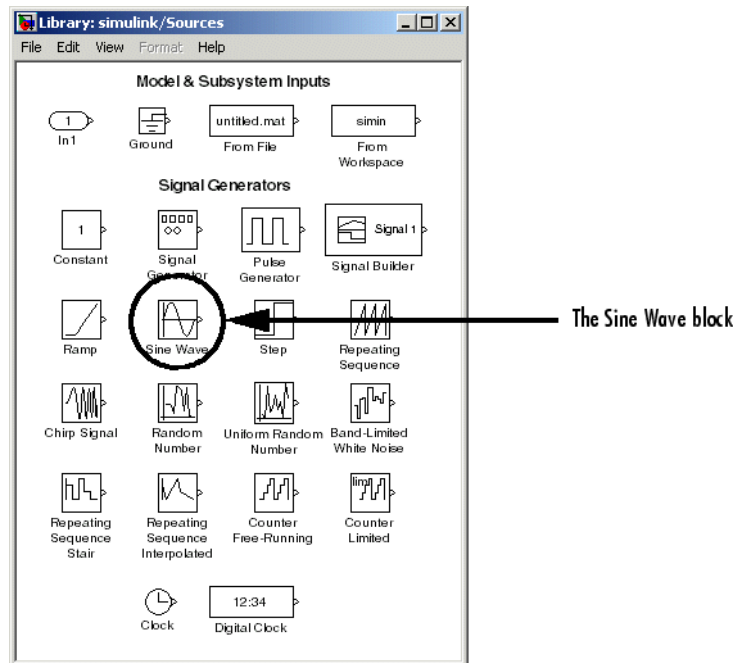
Here is how the Library Browser should look after you have done this.



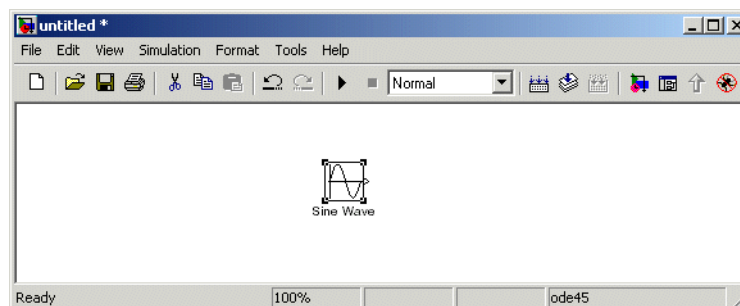
Now drag a copy of the Sine Wave block from the browser and drop it in the model window.

To copy the Sine Wave block from the Sources library window, open the Sources window by double-clicking the Sources icon in the Simulink library window. (On Windows, you can open the Simulink library window by right-clicking the Simulink node in the Library Browser and then clicking the resulting **Open Library** button.)

Simulink displays the Sources library window.



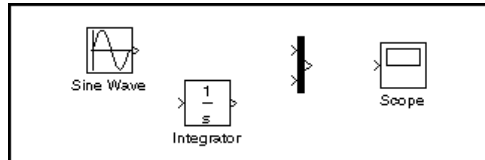
Now drag the Sine Wave block from the Sources window to your model window.



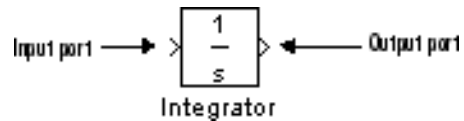
Copy the rest of the blocks in a similar manner from their respective libraries into the model window. You can move a block from one place in the model

window to another by dragging the block. You can move a block a short distance by selecting the block, then pressing the arrow keys.

With all the blocks copied into the model window, the model should look something like this.

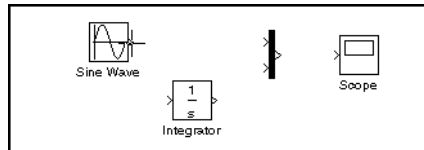


Notice that one or both sides of the blocks have angle brackets. The > symbol pointing out of a block is an *output port*; if the symbol points to a block, it is an *input port*.



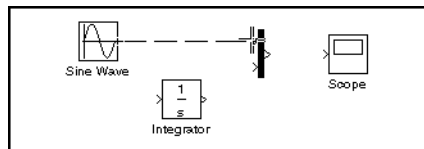
## Connecting the Blocks

Now it's time to connect the blocks. Connect the Sine Wave block to the top input port of the Mux block. Position the pointer over the output port on the right side of the Sine Wave block. Notice that the cursor shape changes to crosshairs.

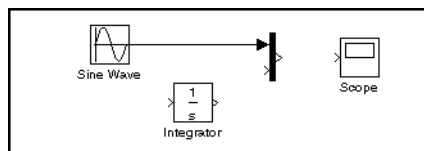


Hold down the mouse button and move the cursor to the top input port of the Mux block.

Notice that the line is dashed while the mouse button is down and that the cursor shape changes to double-lined crosshairs as it approaches the Mux block.



Now release the mouse button. The blocks are connected. You can also connect the line to the block by releasing the mouse button while the pointer is over the block. If you do, the line is connected to the input port closest to the cursor's position.

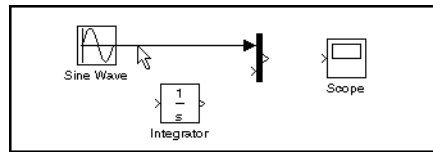


If you look again at the model at the beginning of this section, you'll notice that most of the lines connect output ports of blocks to input ports of other blocks. However, one line connects a *line* to the input port of another block. This line, called a *branch line*, connects the Sine Wave output to the

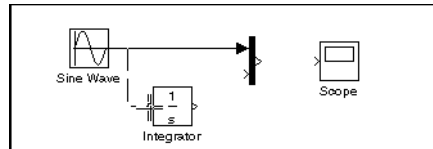
Integrator block, and carries the same signal that passes from the Sine Wave block to the Mux block.

Drawing a branch line is slightly different from drawing the line you just drew. To weld a connection to an existing line, follow these steps:

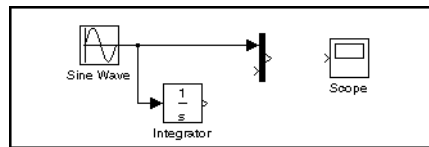
- 1 First, position the pointer *on the line* between the Sine Wave and the Mux block.



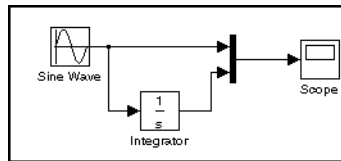
- 2 Press and hold down the **Ctrl** key (or click the right mouse button). Press the mouse button, then drag the pointer to the Integrator block's input port or over the Integrator block itself.



- 3 Release the mouse button. Simulink draws a line between the starting point and the Integrator block's input port.



Finish making block connections. When you're done, your model should look something like this.

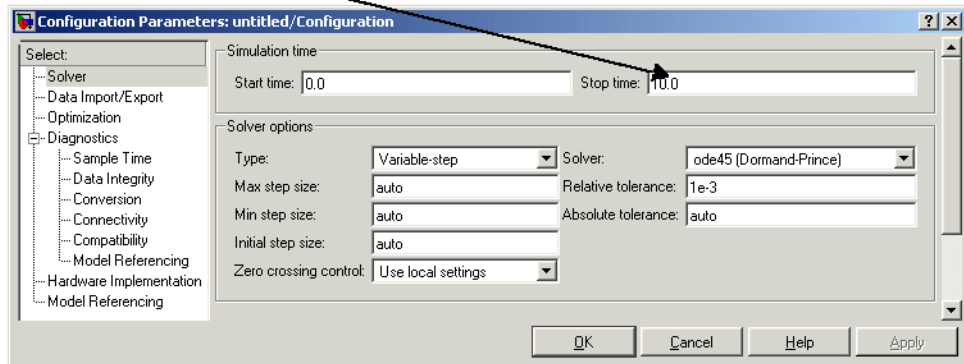




## Configuring the Model

Now set up Simulink to run the simulation for 10 seconds. First, open the **Configuration Parameters** dialog box by choosing **Configuration Parameters** from the **Simulation** menu. On the dialog box that appears, notice that the **Stop time** is set to 10.0 (its default value).

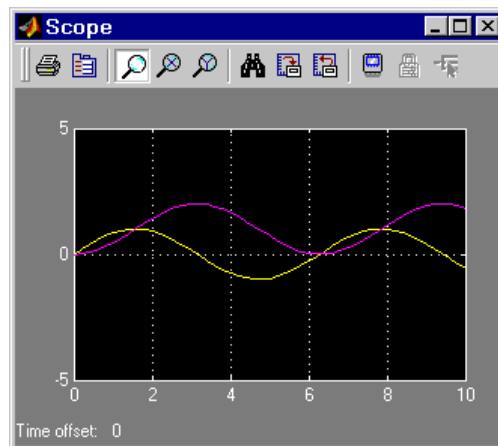
Stop time parameter



Close the **Configuration Parameters** dialog box by clicking the **OK** button. Simulink applies the parameters and closes the dialog box.

## Running the Model

Now double-click the Scope block to open its display window. Finally, choose **Start** from the **Simulation** menu and watch the simulation output on the Scope.



The simulation stops when it reaches the stop time specified in the **Configuration Parameters** dialog box or when you choose **Stop** from the **Simulation** menu or click the **Stop** button on the model window's toolbar (Windows only).

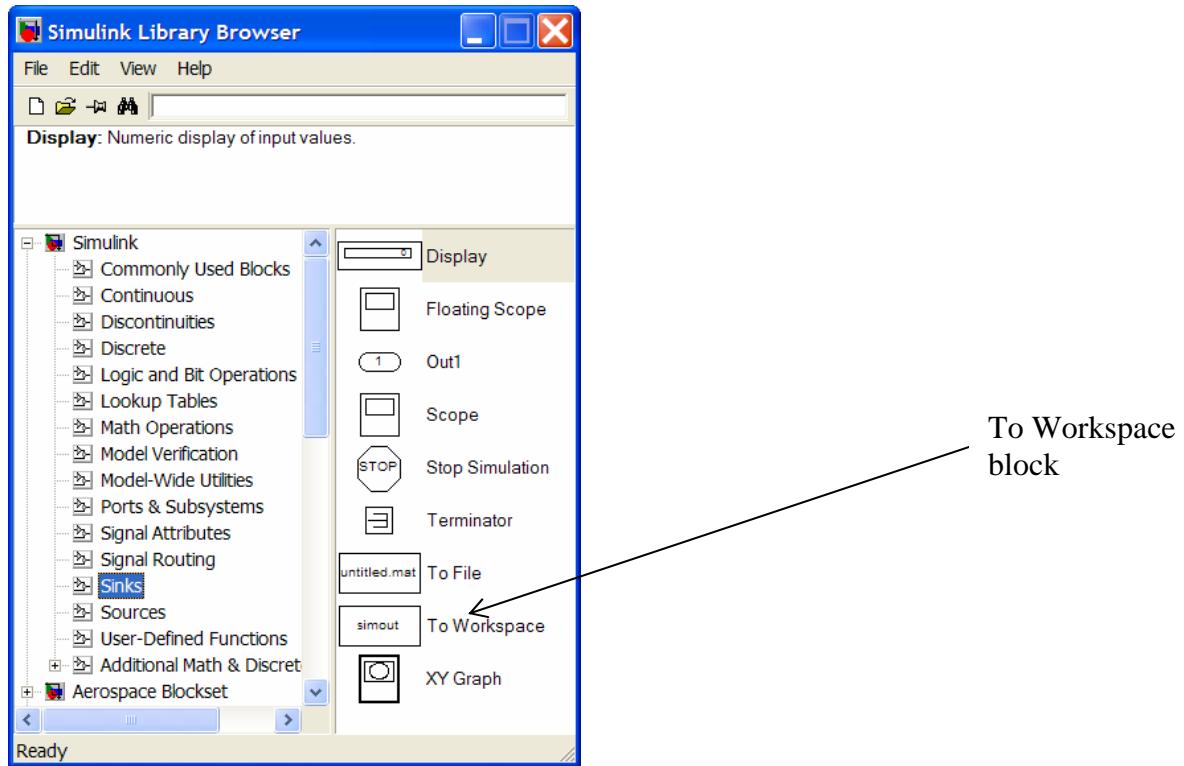
To save this model, choose **Save** from the **File** menu and enter a filename and location. That file contains the description of the model.

To terminate Simulink and MATLAB, choose **Exit MATLAB** (on a Microsoft Windows system) or **Quit MATLAB** (on a UNIX system). You can also enter quit in the MATLAB Command Window. If you want to leave Simulink but not terminate MATLAB, just close all Simulink windows.

This exercise shows you how to perform some commonly used model-building tasks. These and other tasks are described in more detail in "Creating a Model".

## Saving data to Matlab Workspace

If not already open, open the Simulink Library Browser. Drag a copy of the To Workspace block from the browser and drop it in the model window



Open the To Workspace block parameter by double clicking the block. Each Simulink block has a set of configurable parameters.

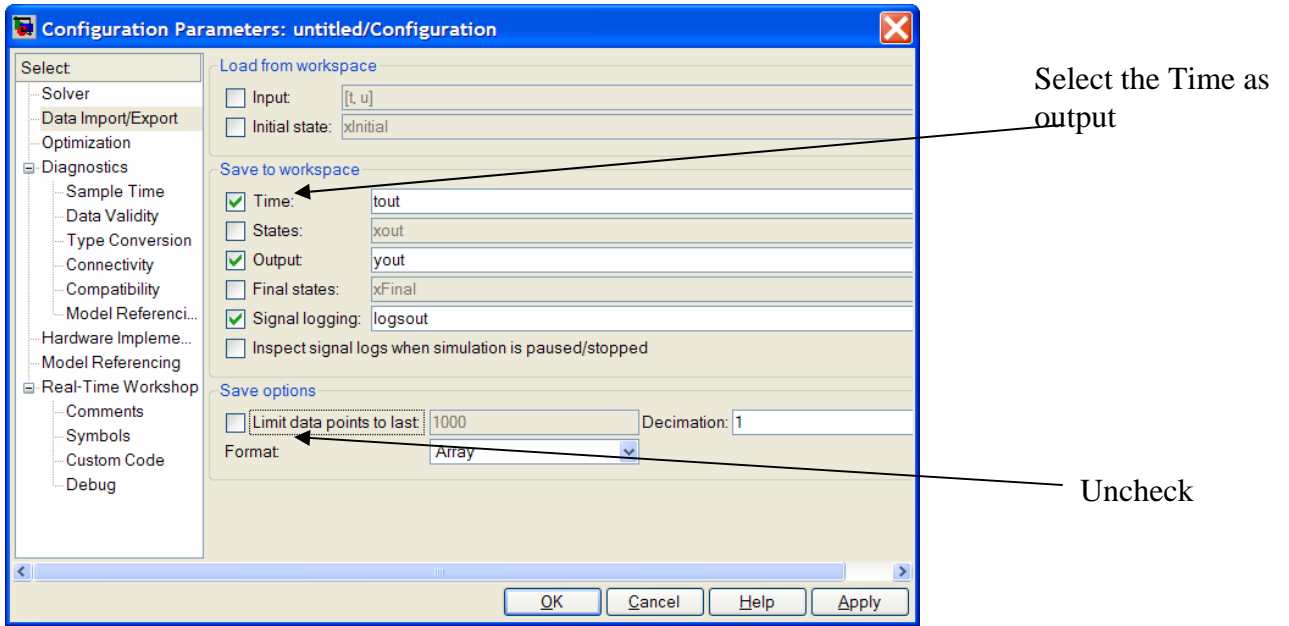
- Write a “good” name as Variable name.
- Select the Save format pop up menu and select Array.

Close the Block Parameter window.

Run the simulation again by choosing **Start** from the **Simulation** menu, or by pushing the black triangle in the graphical menu in the model window, or by typing Ctrl T.

Now check the Matlab workspace and you will find a signal (array) with the name that you choose in the Block parameter. Plot the signal by executing, `<< plot(signalname)` at the Matlab prompt.

To plot the signal as a function of time instead as of only index, do the following. Chose **Configure Parameters** from the **Simulation** menu. Select the Data Import/Export page and verify that the Time box is checked and write a suitable name for the time signal, uncheck the Limit data points to last box.

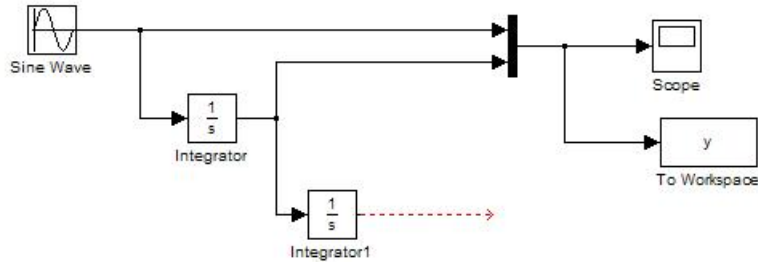


Run the simulation again, check if the time is available in Matlab Workspace and plot the signal again, << plot(time,signal).

## Grouping blocks

The sine model is a very simple model which is easy to understand in terms of blocks and signal routing. For more complex files with many blocks and signals it is important to create a good model structure and signal naming.

Copy the Integrator block by drag and drop with the right hand mouse button. Connect it as shown below, do not connect the output of the second integrator.

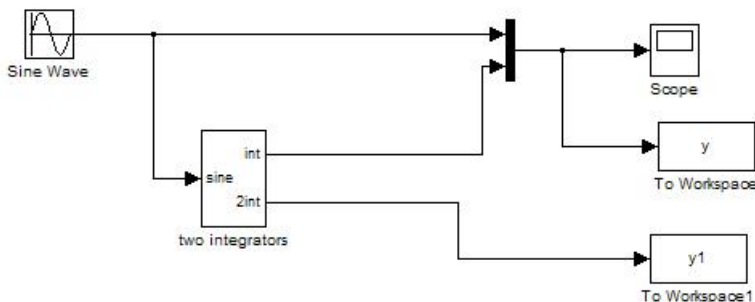


Select both integrator blocks by holding down the right mouse button and at the same time drawing a box around both blocks.

In the **Edit** menu select **Create Subsystem**, reorganize the blocks such that the signal lines do not pass over each other or other blocks.

Click on the Subsystem name and enter a name for the new subsystem. Open the subsystem by double clicking it.

Rename the automatically created input port block and the two output port blocks, close the subsystem and connect a To Workspace block on each of the outputs of the subsystem. It should look something like the figure below. Simulate again and plot both signals.



## Simulating and plotting directly from M-scripts

To speed up development Simulink files can be simulated from the Matlab prompt or even better then from a M-script.

The Matlab command *sim* is used to simulate a Simulink file, type `<< help sim` at the the Matlab prompt for information how the command works.

- Open the Sine Wave block and write symbolic names instead of numerical values for amplitude and frequency.
- Create a new M-script by typing `<< edit` at the Matlab prompt.
- Write a M-script that simulates and plots the output of the first integrator for four different amplitudes and four different frequencies. Combine the frequencies and amplitudes by using a double for loop in the M-script giving a total of 16 simulations.
- Execute the M-script by writing its name at the Matlab prompt.

The result should look something like in the figure below

