

Simulink® 7

Getting Started Guide

MATLAB®
& SIMULINK®

How to Contact MathWorks



www.mathworks.com Web
comp.soft-sys.matlab Newsgroup
www.mathworks.com/contact_TS.html Technical Support



suggest@mathworks.com Product enhancement suggestions
bugs@mathworks.com Bug reports
doc@mathworks.com Documentation error reports
service@mathworks.com Order status, license renewals, passcodes
info@mathworks.com Sales, pricing, and general information



508-647-7000 (Phone)



508-647-7001 (Fax)



The MathWorks, Inc.
3 Apple Hill Drive
Natick, MA 01760-2098

For contact information about worldwide offices, see the MathWorks Web site.

Simulink® Getting Started Guide

© COPYRIGHT 1990–2011 by The MathWorks, Inc.

The software described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from The MathWorks, Inc.

FEDERAL ACQUISITION: This provision applies to all acquisitions of the Program and Documentation by, for, or through the federal government of the United States. By accepting delivery of the Program or Documentation, the government hereby agrees that this software or documentation qualifies as commercial computer software or commercial computer software documentation as such terms are used or defined in FAR 12.212, DFARS Part 227.72, and DFARS 252.227-7014. Accordingly, the terms and conditions of this Agreement and only those rights specified in this Agreement, shall pertain to and govern the use, modification, reproduction, release, performance, display, and disclosure of the Program and Documentation by the federal government (or other entity acquiring for or through the federal government) and shall supersede any conflicting contractual terms or conditions. If this License fails to meet the government's needs or is inconsistent in any respect with federal procurement law, the government agrees to return the Program and Documentation, unused, to The MathWorks, Inc.

Trademarks

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See www.mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

Patents

MathWorks products are protected by one or more U.S. patents. Please see www.mathworks.com/patents for more information.

Revision History

September 2005	Online only	New for Version 6.3 (Release 14SP3)
March 2006	Online only	Revised for Simulink 6.4 (Release 2006a)
September 2006	Online only	Revised for Simulink 6.5 (Release 2006b)
March 2007	First printing	Revised for Simulink 6.6 (Release 2007a)
September 2007	Second printing	Revised for Simulink 7.0 (Release 2007b)
March 2008	Third printing	Revised for Simulink 7.1 (Release 2008a)
October 2008	Fourth printing	Revised for Simulink 7.2 (Release 2008b)
March 2009	Fifth printing	Revised for Simulink 7.3 (Release 2009a)
September 2009	Online only	Revised for Simulink 7.4 (Release 2009b)
March 2010	Online only	Revised for Simulink 7.5 (Release 2010a)
September 2010	Online only	Revised for Simulink 7.6 (Release 2010b)
April 2011	Online only	Revised for Simulink 7.7 (Release 2011a)

Introduction

1

Product Overview	1-2
Overview	1-2
Tool for Model-Based Design	1-3
Tool for Simulation	1-4
Tool for Analysis	1-4
How Simulink Software Interacts with the MATLAB Environment	1-4
What Is Model-Based Design?	1-5
Model-Based Design	1-5
Modeling Process	1-6
Related Products	1-9

Simulink Software Basics

2

Starting Simulink Software	2-2
Opening the Simulink Library Browser	2-2
Creating a New Model	2-3
Opening an Existing Model	2-5
Simulink User Interface	2-6
Simulink Library Browser	2-6
Simulink Model Window	2-9
Getting Help with Simulink Software	2-10
Simulink Online Help	2-10
Simulink Demo Models	2-11
Web Site Resources	2-13

Creating a Simulink Model

3

Overview of a Simple Model	3-2
Creating a Simple Model	3-3
Creating a New Model	3-3
Adding Blocks to the Model	3-4
Moving Blocks in the Model Window	3-6
Connecting Blocks in a Simple Model	3-7
Block Connections in the Model Window	3-7
Drawing Lines Between Blocks	3-7
Drawing a Branch Line	3-9
Simulating the Simple Model	3-11
Setting Simulation Options	3-11
Running the Simulation and Observing Results	3-13

Modeling a Dynamic Control System

4

Overview of a Demo Model	4-2
Understanding the Demo Model	4-3
Opening the Demo Model	4-3
Anatomy of the Demo Model	4-5
Subsystems in the Demo Model	4-6
Masking Subsystems	4-9
Simulating the Demo Model	4-11
Running the Simulation	4-11
Modifying Simulation Parameters	4-13
Importing Data from the MATLAB Workspace	4-20
Exporting Simulation Data to the MATLAB Workspace ..	4-25

Completing a Basic Simulation Workflow

5

Overview	5-2
Importing Data from a Microsoft® Excel Spreadsheet ..	5-4
Configuring the Model	5-9
For Logging Output Signals	5-9
Simulating the Model	5-10
Comparing Signals in the Simulation Data Inspector ..	5-13
Creating, Saving, and Printing a Figure in Simulation	
Data Inspector	5-15
Creating a Figure from a Plot	5-15
Saving and Printing a Figure	5-16

Index

Introduction

- “Product Overview” on page 1-2
- “What Is Model-Based Design?” on page 1-5
- “Related Products” on page 1-9

Product Overview

In this section...
“Overview” on page 1-2
“Tool for Model-Based Design” on page 1-3
“Tool for Simulation” on page 1-4
“Tool for Analysis” on page 1-4
“How Simulink Software Interacts with the MATLAB Environment” on page 1-4

Overview

Simulink® software models, simulates, and analyzes dynamic systems. It enables you to pose a question about a system, model the system, and see what happens.

With Simulink, you can easily build models from scratch, or modify existing models to meet your needs. Simulink supports linear and nonlinear systems, modeled in continuous time, sampled time, or a hybrid of the two. Systems can also be multirate — having different parts that are sampled or updated at different rates.

Thousands of scientists and engineers around the world use Simulink to model and solve real problems in a variety of industries, including:

- Aerospace and Defense
- Automotive
- Communications
- Electronics and Signal Processing
- Medical Instrumentation

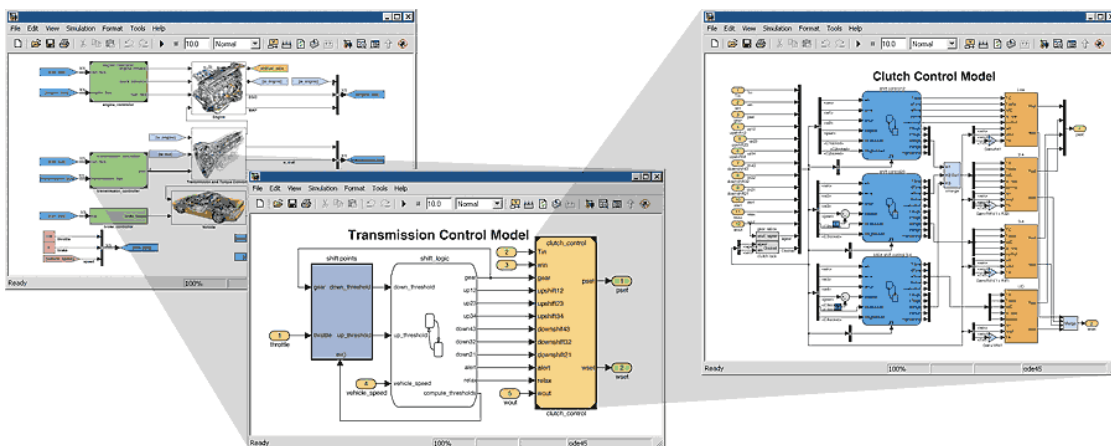
Tool for Model-Based Design

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 laboratory for modeling and analyzing systems that would not be possible or practical otherwise.

Whether you are interested in the behavior of an automotive clutch system, the flutter of an airplane wing, or the effect of the monetary supply on the economy, Simulink provides you with the tools to model and simulate almost any real-world problem. Simulink also provides demos that model a wide variety of real-world phenomena (see “Simulink Demo Models” on page 2-11).

Simulink provides a graphical user interface (GUI) for building models as block diagrams, allowing you to draw models as you would with pencil and paper. Simulink also includes a comprehensive block library of sinks, sources, linear and nonlinear components, and connectors. If these blocks do not meet your needs, however, you can also create your own blocks. The interactive graphical environment simplifies the modeling process, eliminating the need to formulate differential and difference equations in a language or program.

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 see increasing levels of model detail. This approach provides insight into how a model is organized and how its parts interact.



Tool for Simulation

After you define a model, you can simulate it, using a choice of mathematical integration methods, either from the Simulink menus or by entering commands in the MATLAB® Command Window. The menus are convenient for interactive work, while the command line is useful for running a batch of simulations (for example, if you are doing Monte Carlo simulations or want to apply a parameter across a range of values).

Using scopes and other display blocks, you can see the simulation results while the simulation runs. You can then 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.

Tool for Analysis

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. Because MATLAB and Simulink are integrated, you can simulate, analyze, and revise your models in either environment at any point.

How Simulink Software Interacts with the MATLAB Environment

Simulink software is tightly integrated with the MATLAB environment. It requires MATLAB to run, depending on it to define and evaluate model and block parameters. Simulink can also utilize many MATLAB features. For example, Simulink can use the MATLAB environment to:

- Define model inputs.
- Store model outputs for analysis and visualization.
- Perform functions within a model, through integrated calls to MATLAB operators and functions.

What Is Model-Based Design?

In this section...
“Model-Based Design” on page 1-5
“Modeling Process” on page 1-6

Model-Based Design

Model-Based Design is a process that enables faster, more cost-effective development of dynamic systems, including control systems, signal processing, and communications systems. In Model-Based Design, a system model is at the center of the development process, from requirements development, through design, implementation, and testing. The model is an executable specification that is continually refined throughout the development process. After model development, simulation shows whether the model works correctly.

When software and hardware implementation requirements are included, such as fixed-point and timing behavior, you can automatically generate code for embedded deployment and create test benches for system verification, saving time and avoiding the introduction of manually coded errors.

Model-Based Design allows you to improve efficiency by:

- Using a common design environment across project teams
- Linking designs directly to requirements
- Integrating testing with design to continuously identify and correct errors
- Refining algorithms through multidomain simulation
- Automatically generating embedded software code
- Developing and reusing test suites
- Automatically generating documentation
- Reusing designs to deploy systems across multiple processors and hardware targets

Modeling Process

There are six steps to modeling any system:

- 1** Defining the System
- 2** Identifying System Components
- 3** Modeling the System with Equations
- 4** Building the Simulink Block Diagram
- 5** Running the Simulation
- 6** Validating the Simulation Results

You perform the first three steps of this process outside of the Simulink software before you begin building your model.

Defining the System

The first step in modeling a dynamic system is to fully define the system. If you are modeling a large system that can be broken into parts, you should model each subcomponent on its own. Then, after building each component, you can integrate them into a complete model of the system.

For example, the demo model used later in this guide models the heating system of a house. This system can be broken down into three main parts:

- Heater subsystem
- Thermostat subsystem
- Thermodynamic model subsystem

The most effective way to build a model of this system is to consider each of these subsystems independently.

Identifying System Components

The second step in the modeling process is to identify the system components. Three types of components define a system:

- **Parameters** — System values that remain constant unless you change them
- **States** — Variables in the system that change over time
- **Signals** — Input and output values that change dynamically during the simulation

In Simulink, parameters and states are represented by blocks, while signals are represented by the lines that connect blocks.

For each subsystem that you identified, ask yourself the following questions:

- How many input signals does the subsystem have?
- How many output signals does the subsystem have?
- How many states (variables) does the subsystem have?
- What are the parameters (constants) in the subsystem?
- Are there any intermediate (internal) signals in the subsystem?

Once you have answered these questions, you should have a comprehensive list of the system components, and are ready to begin modeling the system.

Modeling the System with Equations

The third step in modeling a system is to formulate the mathematical equations that describe the system.

For each subsystem, use the list of system components you identified to describe the system mathematically. Your model may include:

- Algebraic equations
- Logical equations
- Differential equations, for continuous systems
- Difference equations, for discrete systems

You use these equations to create the block diagram in Simulink.

Building the Simulink Block Diagram

After you have defined the mathematical equations that describe each subsystem, you can begin building a block diagram of your model in Simulink.

Build the block diagram for each of your subcomponents separately. After you have modeled each subcomponent, you can then integrate them into a complete model of the system.

See “Creating a Simple Model” on page 3-3 for more information on building the block diagram.

Running the Simulation

After you build the Simulink block diagram, you can simulate the model and analyze the results.

Simulink allows you to interactively define system inputs, simulate the model, and observe changes in behavior. This allows you to quickly evaluate your model.

See “Simulating the Simple Model” on page 3-11 for more information on running a simulation.

Validating the Simulation Results

Finally, you must validate that the model accurately represents the physical characteristics of the system.

You can use the linearization and trimming tools available from the MATLAB command line, plus the many tools in MATLAB and its application toolboxes to analyze and validate your model.

Related Products

Many additional MathWorks® products extend the capabilities of Simulink software. For information about these related products, see <http://www.mathworks.com/products/simulink/>.

Simulink Software Basics

- “Starting Simulink Software” on page 2-2
- “Simulink User Interface” on page 2-6
- “Getting Help with Simulink Software” on page 2-10

Starting Simulink Software

In this section...
“Opening the Simulink Library Browser” on page 2-2
“Creating a New Model” on page 2-3
“Opening an Existing Model” on page 2-5

Opening the Simulink Library Browser

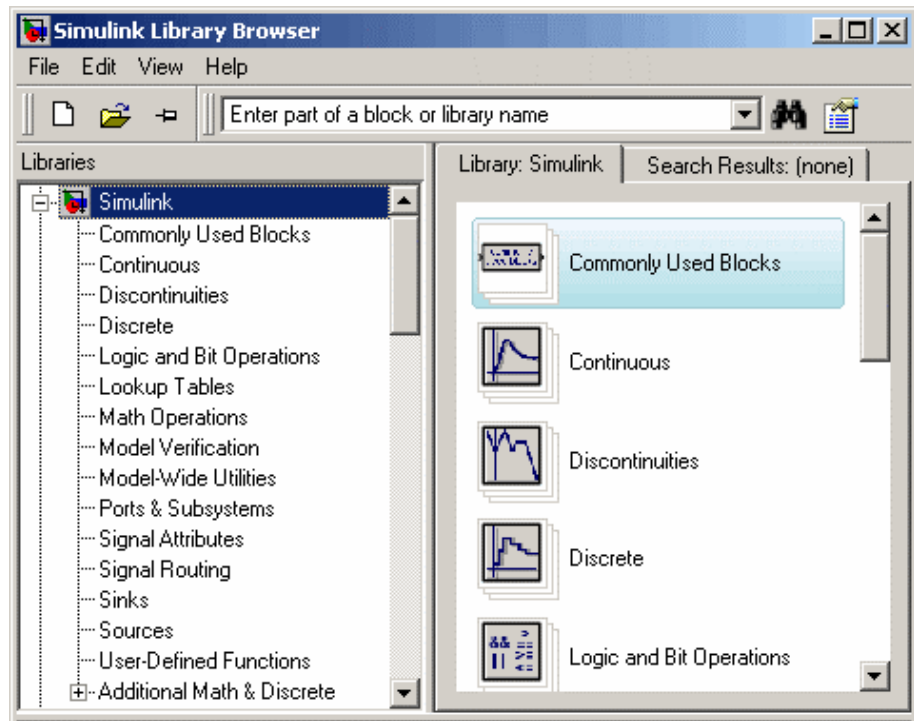
Your MATLAB environment must be running before you can open the Simulink Library Browser. You start Simulink software from within MATLAB.

1 Start MATLAB. For more information, see “Starting a MATLAB Session” in the *MATLAB Getting Started Guide*.


2 In the MATLAB Command Window, enter

```
simulink
```

The Simulink Library Browser opens.



Note You can also open the Simulink Library Browser:

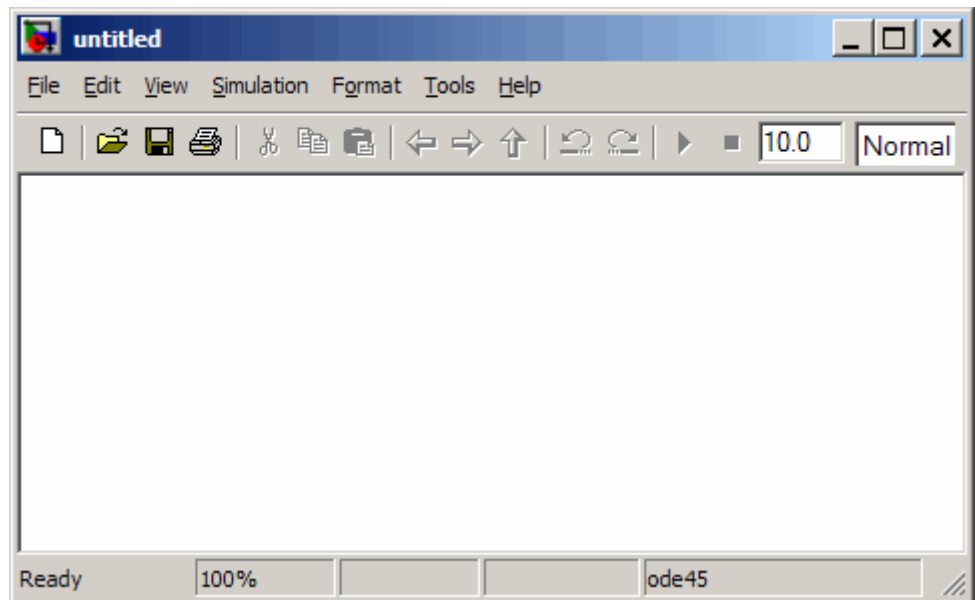
- From the MATLAB toolbar, click the Simulink icon 
- From the MATLAB task bar, click the **Start** button, and then select **Simulink > Library Browser**

Creating a New Model

You can create a new model from the Simulink Library Browser.

- 1 From the Simulink Library Browser menu, select **File > New > Model**.

A new model opens in the Simulink model window.



2 Select **File > Save**.

The Save As dialog box opens.

3 In the **File name** box, enter a name for your model, and then click **Save**.

The software saves your model with the file extension `.mdl`.

Opening an Existing Model

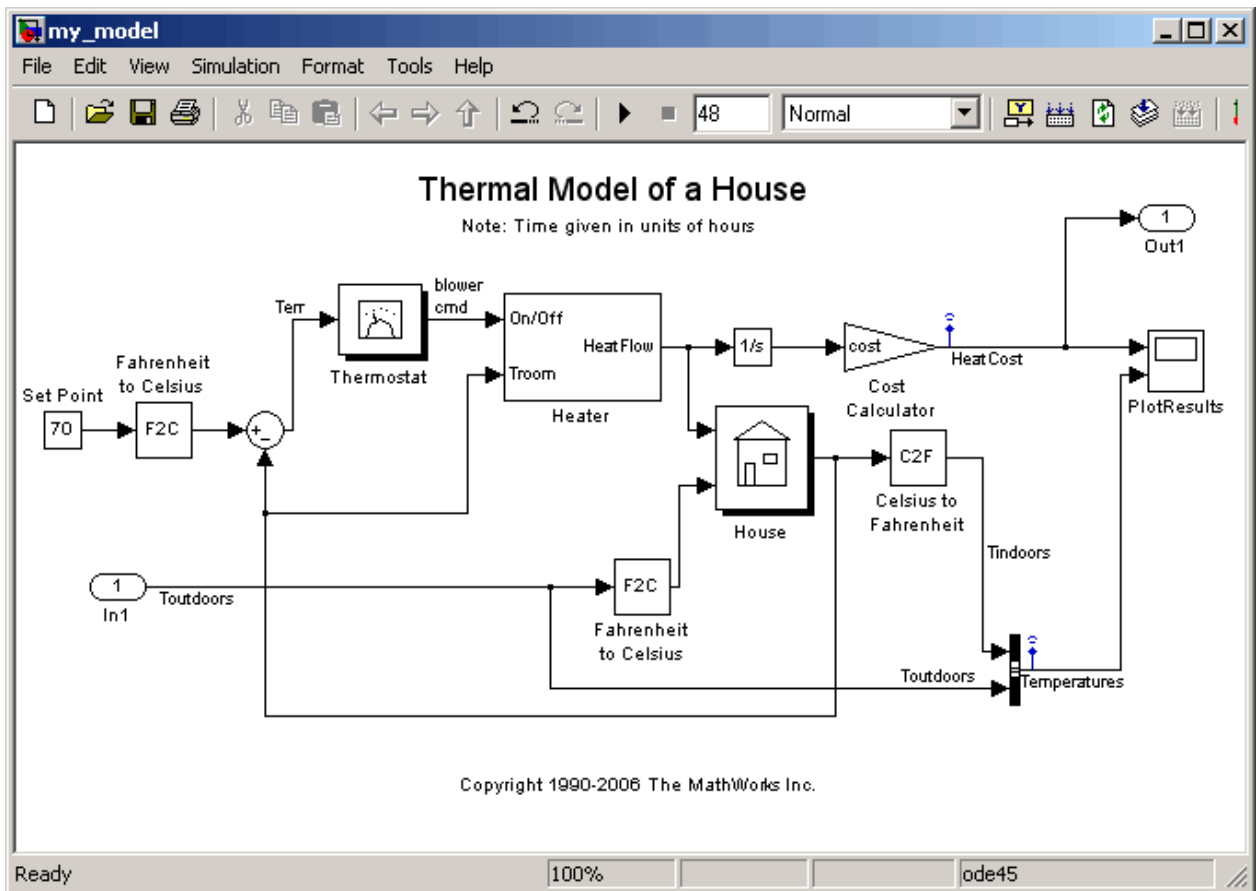
You can open an existing Simulink model from the Simulink Library Browser.

- 1 From the Simulink Library Browser menu, select **File > Open**.

The Open dialog box opens.

- 2 Select the model (.mdl file) you want to open, and then click **Open**.

The selected model opens in the Simulink model window.

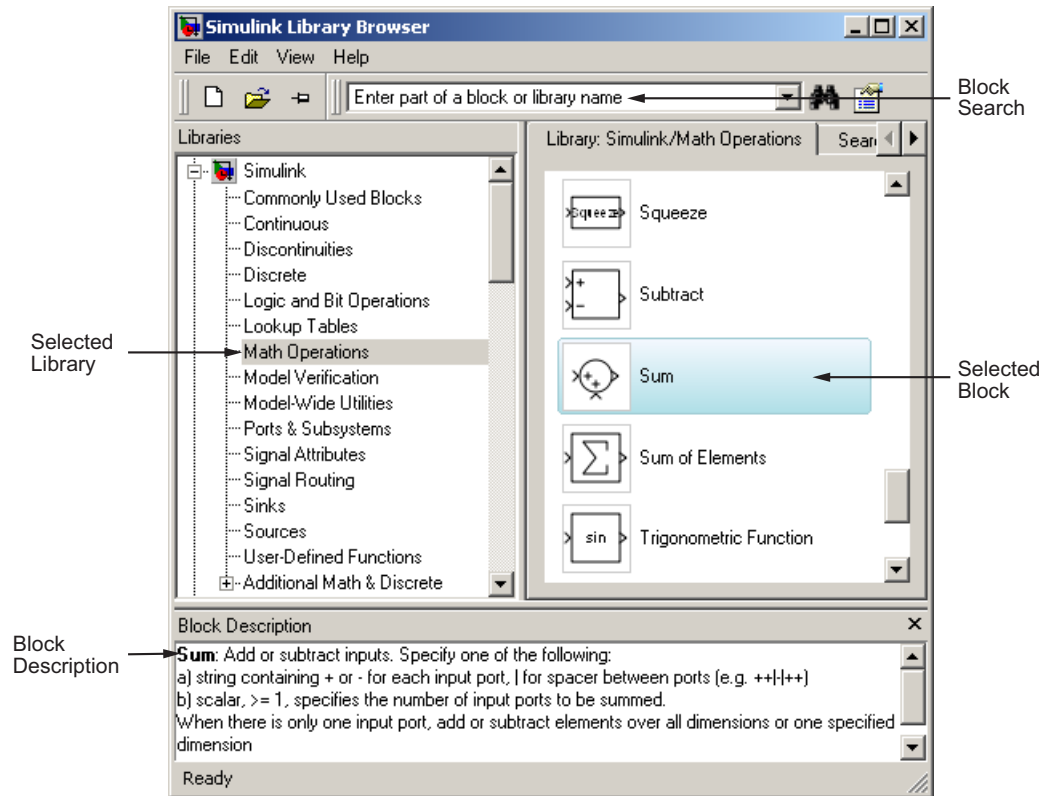


Simulink User Interface

In this section...
“Simulink Library Browser” on page 2-6
“Simulink Model Window” on page 2-9

Simulink Library Browser


The Simulink Library Browser displays the block libraries installed on your computer. You build models by copying blocks from a library into a model window.



Simulink® Library Browser

Tips for Using the Library Browser

When using the Library Browser, note the following:

- View the blocks in a library by selecting the library name in the left side of the Library Browser, or by double-clicking the library icon in the right side.
- When you select a block, a description of that block appears at the bottom of the browser.
- For more information on a block, select the block, and then select **Help > Help on the Selected Block** to display the help page for the block.
- View the parameters for a block by right-clicking the block, and then selecting **Block Parameters**.
- Search for a specific block by entering the name of the block in the block search field, and then clicking the Find block icon .

Standard Block Libraries

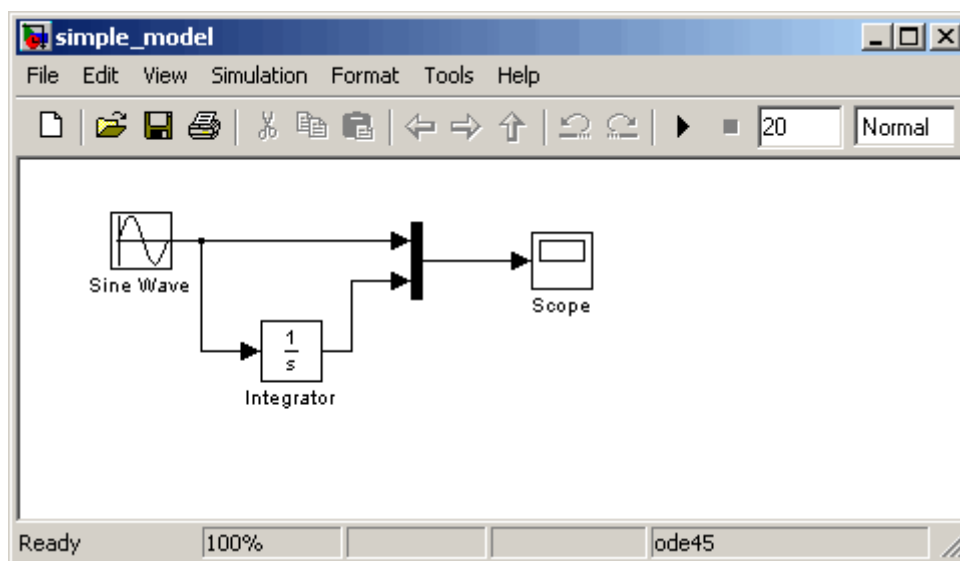
Simulink software includes 16 standard block libraries. The following table describes each of these libraries.

Block Library	Description
Commonly Used Blocks	Contains a group of the most commonly used blocks, such as the Constant , In1 , Out1 , Scope , and Sum blocks. Each of the blocks in this library are also included in other libraries.
Continuous	Contains blocks that model linear functions, such as the Derivative and Integrator blocks.
Discontinuities	Contains blocks with outputs that are discontinuous functions of their inputs, such as the Saturation block.
Discrete	Contains blocks that represent discrete time functions, such as the Unit Delay block.
Logic and Bit Operations	Contains blocks that perform logic or bit operations, such as the Logical Operator and Relational Operator blocks.

Block Library	Description
LookUp Tables	Contains blocks that use lookup tables to determine their outputs from their inputs, such as the Cosine and Sine blocks.
Math Operations	Contains blocks that perform mathematical and logical functions, such as the Gain , Product , and Sum blocks.
Model Verification	Contains blocks that enable you to create self-validating models, such as the Check Input Resolution block.
Model-Wide Utilities	Contains blocks that provide information about the model, such as the Model Info block.
Ports & Subsystems	Contains blocks that allow you to create subsystems, such as the In1 , Out1 , and Subsystem blocks.
Signal Attributes	Contains blocks that modify the attributes of signals, such as the Data Type Conversion block.
Signal Routing	Contains blocks that route signals from one point in a block diagram to another, such as the Mux and Switch blocks.
Sinks	Contains blocks that display or export output, such as the Out1 and Scope blocks.
Sources	Contains blocks that generate or import system inputs, such as the Constant , In1 , and Sine Wave blocks.
User-Defined Functions	Contains blocks that allow you to define custom functions, such as the MATLAB Function block.
Additional Math & Discrete	Contains two additional libraries for mathematical and discrete function blocks.

Simulink Model Window

The Simulink model window contains a block diagram of your model. You build models in the model window by arranging blocks logically, setting the parameters for each block, and then connecting the blocks with signal lines.



The model window also allows you to:

- Set configuration parameters for the model, including the start and stop time, type of solver to use, and data import/export settings.
- Start and stop a simulation of the model.
- Save the model.
- Print the block diagram.

Getting Help with Simulink Software

In this section...
“Simulink Online Help” on page 2-10
“Simulink Demo Models” on page 2-11
“Web Site Resources” on page 2-13

Simulink Online Help

Simulink software provides comprehensive online help that describes features, blocks, and functions, and provides detailed procedures for common tasks. The help includes online versions of all Simulink documentation, including:

- *Simulink Getting Started Guide* (this guide)
- *Simulink User’s Guide*
- *Simulink Reference*
- *Writing S-Functions*
- *Simulink Release Notes*

You access the online help from the **Help** menu of the Simulink Library Browser or the Simulink model window.

To access the online help:

- From the Simulink Library Browser, select **Help > Simulink Help**.
- From the Simulink model window, select **Help > Using Simulink**.

Note To quickly access the help page for a specific block, right-click the block and select **Help**.

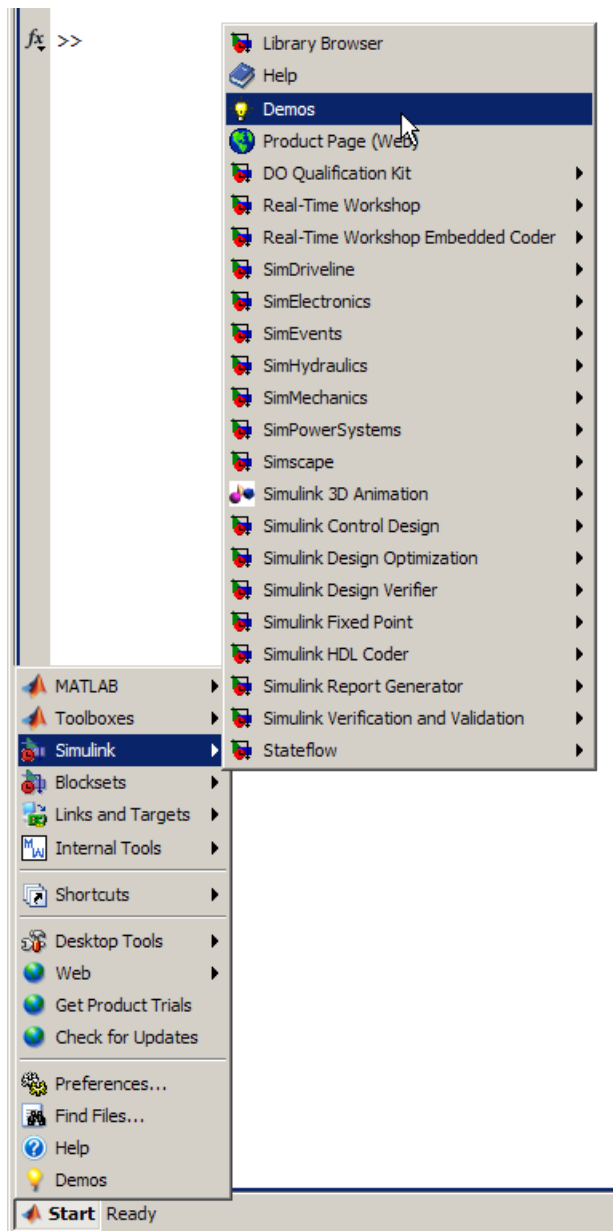
Simulink Demo Models

Simulink software provides a variety of demo models that illustrate key modeling concepts and Simulink features. You can access these demos from the MATLAB Command Window.

- 1 From the MATLAB task bar, click the **Start** button.

The **Start** menu appears.

- 2 From the Start menu, select **Simulink > Demos**.



- 3 From the Simulink demos page in the MATLAB Help browser, click the demo model you want to open.

Web Site Resources

You can access additional Simulink resources on the MathWorks Web site, including Simulink related books, prerecorded webinars, and technical support.

To access the Simulink product page, go to

<http://www.mathworks.com/products/simulink>

Creating a Simulink Model

- “Overview of a Simple Model” on page 3-2
- “Creating a Simple Model” on page 3-3
- “Connecting Blocks in a Simple Model” on page 3-7
- “Simulating the Simple Model” on page 3-11

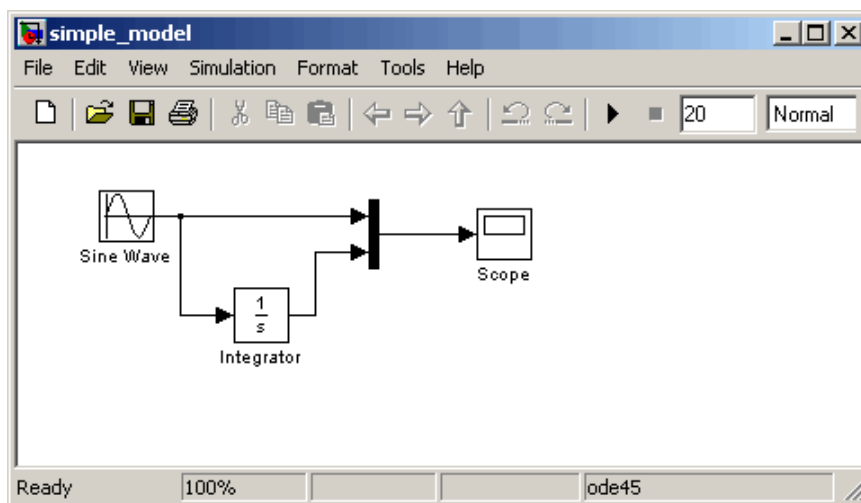
Overview of a Simple Model

You use Simulink software to model dynamic systems and simulate the behavior of the models. The basic techniques you use to create a simple model with Simulink are the same techniques you use for more complex models.

To create a simple model, you need four blocks:

- **Sine Wave** — Generates an input signal for the model.
- **Integrator** — Processes the input signal.
- **Mux** — Multiplexes the input signal and processed signal into a single signal.
- **Scope** — Visualizes the signals in the model.

After the blocks are connected, they model a system that integrates a sine wave signal and displays the result along with the original signal.



You can build this simple model yourself, starting with “Creating a New Model” on page 3-3. The instructions for creating and simulating the simple model are brief. For more detailed information about each task, see the Simulink User’s Guide.

Creating a Simple Model

In this section...

“Creating a New Model” on page 3-3

“Adding Blocks to the Model” on page 3-4

“Moving Blocks in the Model Window” on page 3-6

Creating a New Model

Before you can begin building your model, you must first start Simulink and then create an empty model.

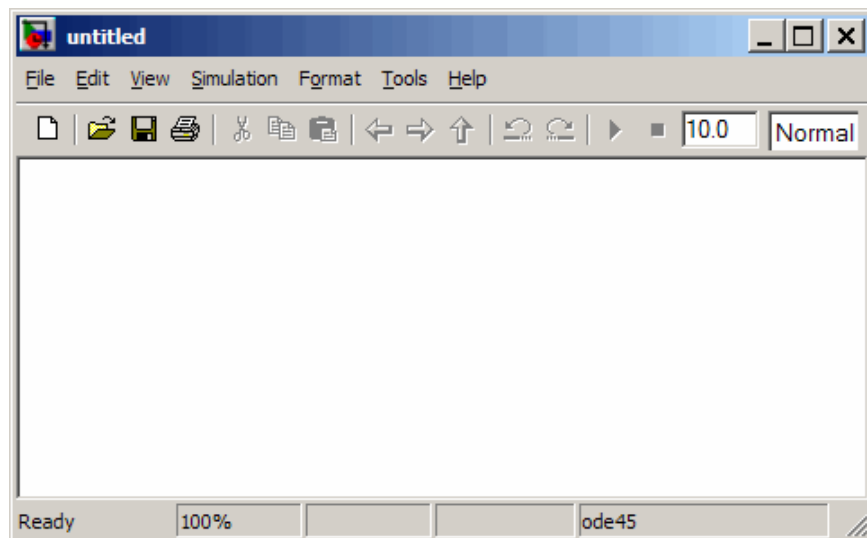
- 1 If Simulink is not running, in the MATLAB Command Window, enter

```
simulink
```

The Simulink Library Browser opens.

- 2 From the Simulink Library Browser menu, select **File > New > Model**.

The Simulink model window opens with an empty model.



3 Select **File > Save**.

The Save As dialog box opens.

4 In the **File name** box, enter a name of your model, and then click **Save**. For example, enter `simple_model`.

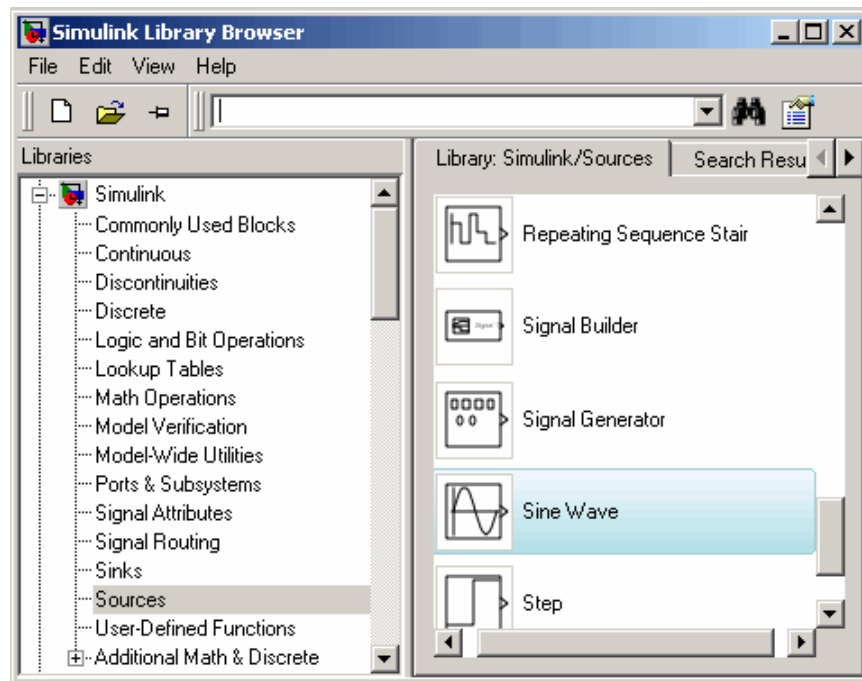
The software saves your model with the filename `simple_model.mdl`.

Adding Blocks to the Model

To create a model, you begin by copying blocks from the Simulink Library Browser to the Simulink model window. For a description of the blocks in this example, see “Overview of a Simple Model” on page 3-2.

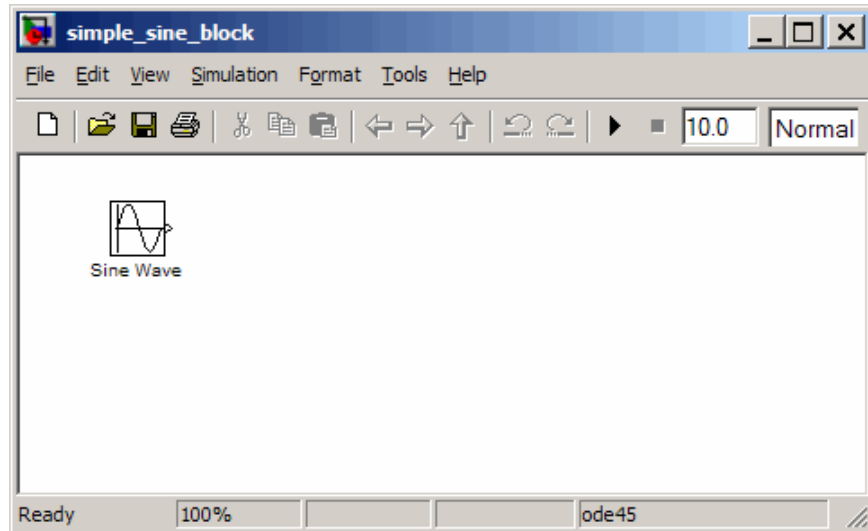
1 In the Simulink Library Browser, select the Sources library.

The Simulink Library Browser displays the Sources library in the right pane.



- 2** Select the **Sine Wave** block, and then drag it to the model window.

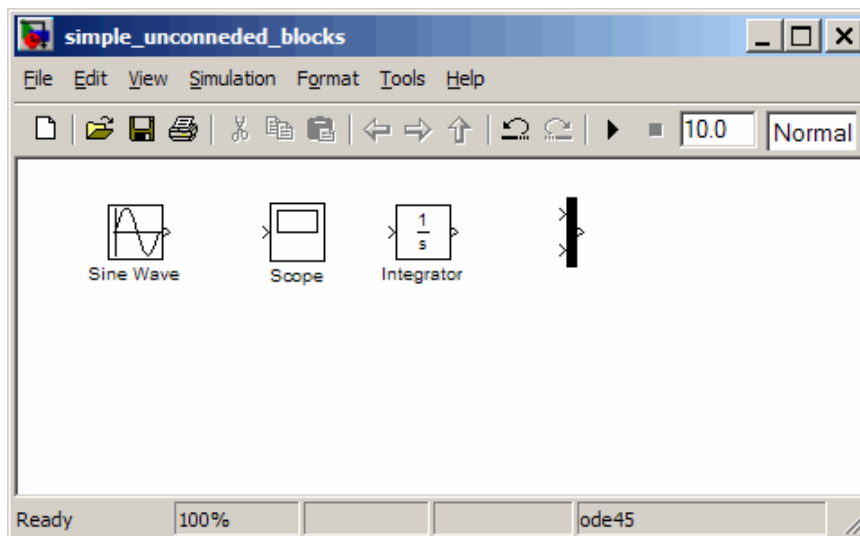
A copy of the Sine Wave block appears in the model window.



- 3** Add the following blocks to your model in the same way you added the **Sign Wave** block.

Library	Block
Sinks	Scope
Continuous	Integrator
Signal Routing	Mux

Your model now has the blocks for a simple model.



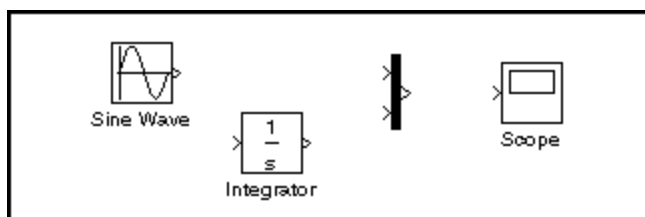
Moving Blocks in the Model Window

Before you connect the blocks in your model, you should arrange them logically to make the signal connections as straightforward as possible.

To move a block in the model window, you can either:

- Click and drag the block.
- Select the block, and then press the arrow keys on the keyboard.

Arrange the blocks in the model to look like the following figure.



Connecting Blocks in a Simple Model

In this section...

“Block Connections in the Model Window” on page 3-7

“Drawing Lines Between Blocks” on page 3-7

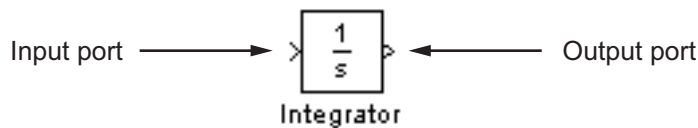
“Drawing a Branch Line” on page 3-9

Block Connections in the Model Window

After you add blocks to the model window, you must connect them. The connecting lines represent the signals within the model.

Most blocks have angle brackets on one or both sides. These angle brackets represent input and output ports:

- The > symbol pointing into a block is an *input port*.
- The > symbol pointing out of a block is an *output port*.

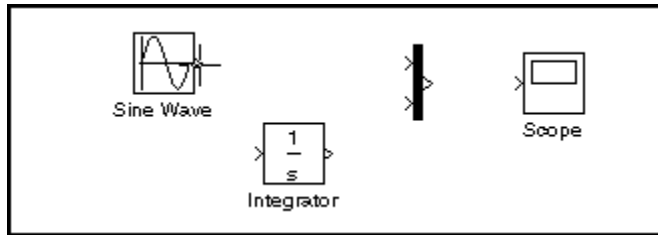


Drawing Lines Between Blocks

After you add blocks to your model, you can connect the blocks by drawing lines between output ports and input ports. For how to add blocks to the model in this example, see “Adding Blocks to the Model” on page 3-4.

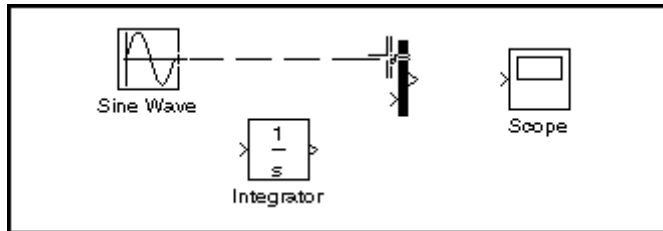
- 1 Position the mouse pointer over the output port on the right side of the Sine Wave block.

The pointer changes to a cross hairs (+) shape while over the port.



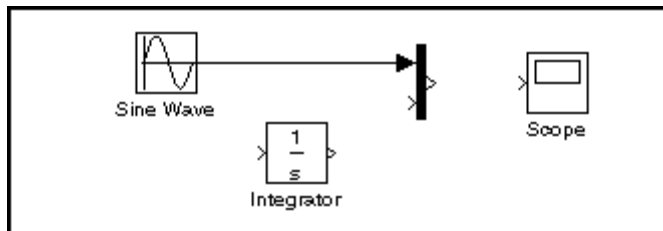
- 2 Drag a line from the output port to the top input port of the Mux block.

While you hold the mouse button down, the connecting line is dashed. The pointer changes to a double-lined cross hairs as it approaches the input port of the Mux block.



- 3 Release the mouse button over the output port.

The blocks connect with an arrow that indicates the direction of signal flow.



- 4 Drag a line from the output port of the Integrator block to the bottom input port on the Mux block.

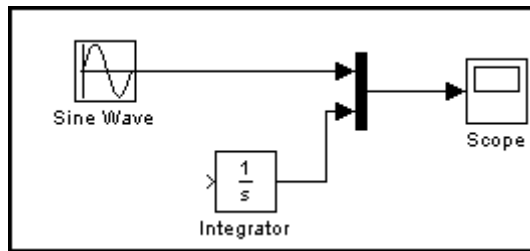
The Integrator block connects to the Mux block with a signal line.

- 5 Select the Mux block, hold down the **Ctrl** key, and then select the Scope block.

A line automatically draws between the blocks to connect them.

Note The **Ctrl+click** shortcut is useful when you are connecting widely separated blocks, or when working with complex models.

The model should now look similar to the following figure.

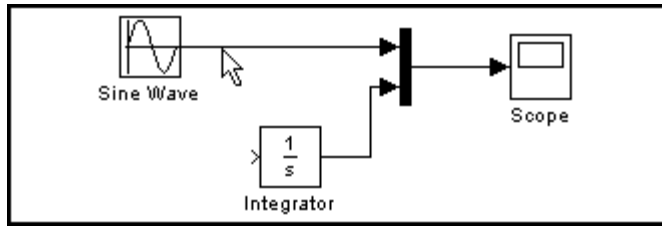


Drawing a Branch Line

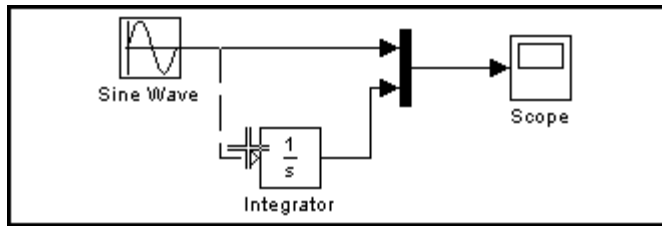
The model is almost complete, but one connection is missing. To finish the model, you must connect the Sine Wave block to the Integrator block.

This final connection is somewhat different from the other three connections, which all connect output ports to input ports. Because the output port of the Sine Wave block already has a connection, you must connect this existing line to the input port of the Integrator block. The new line, called a *branch line*, carries the same signal that passes from the Sine Wave block to the Mux block.

- 1 Position the mouse pointer *on the line* between the Sine Wave and the Mux block.



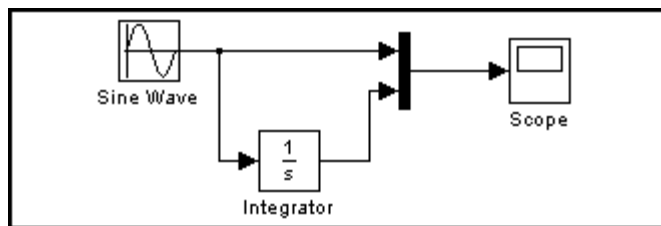
- 2 Hold down the **Ctrl** key, and then drag a line to the input port of the Integrator block input port.



This step adds a connection to the existing line and draws a line between the connection and the input port of the Integrator block.

- 3 From the **File** menu, click **Save**.

The model is now complete. It should look similar to the following figure.



Now you can simulate the model as described in “Simulating the Simple Model” on page 3-11.

Simulating the Simple Model

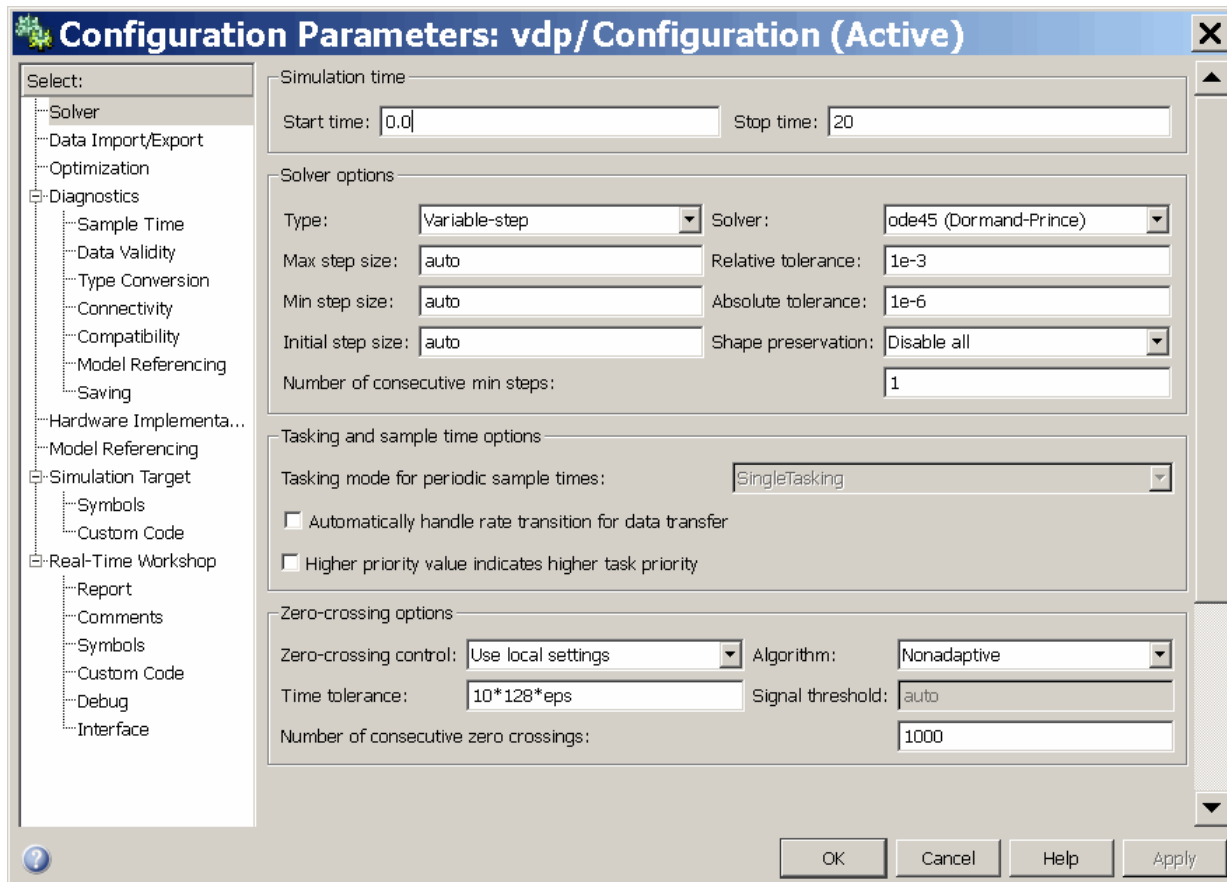
In this section...
“Setting Simulation Options” on page 3-11
“Running the Simulation and Observing Results” on page 3-13

Setting Simulation Options

After you finish building a model, you can simulate and visualize the results, but before you simulate a model, you have to set simulation options. You specify options, such as the stop time and solver, using the Configuration Parameters dialog box. For how to build the model in this example, see “Creating a Simple Model” on page 3-3.

- 1 In the Simulink model window, select **Simulation > Configuration Parameters**.

The Configuration Parameters dialog box opens.



2 In the **Stop time** field, enter 20, and then Click **OK**.

The software updates the parameter values with your changes and closes the Configuration Parameters dialog box.



Note For more information about Simulink configuration parameters, see “Configuration Parameters Dialog Box”.

Running the Simulation and Observing Results

After entering your configuration parameter changes, you are ready to simulate the simple model and visualize the simulation results.

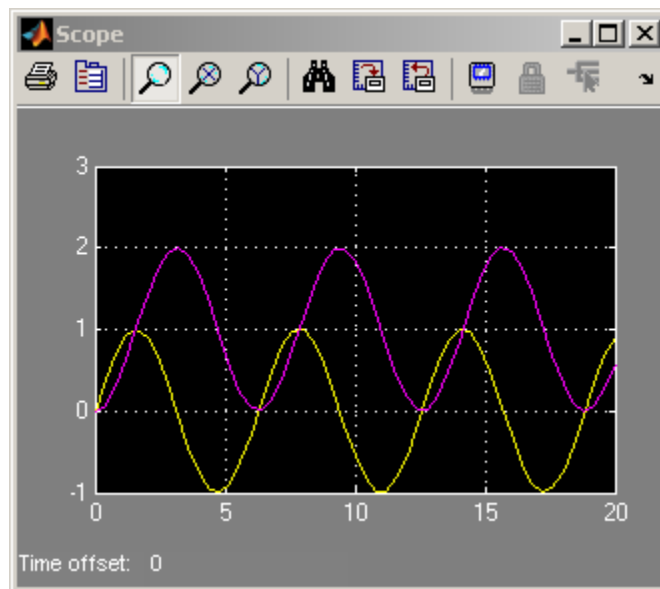
- 1 In the Simulink model window, select **Simulation > Start**.

The simulation runs, stopping when it reaches the stop time specified in the Configuration Parameters dialog box.

Tip On computers running the Microsoft® Windows® operating system, you can click the **Start simulation** button  and **Stop simulation** button  in the model window toolbar to start and stop a simulation.

- 2 Double-click the **Scope** block.

The Scope window opens and displays the simulation results.



The plot shows a sine wave signal with the resulting cosine wave signal from the Integrator block.

3 Select **File > Close**.

The model window closes with your changes to the configuration parameters.

Modeling a Dynamic Control System

- “Overview of a Demo Model” on page 4-2
- “Understanding the Demo Model” on page 4-3
- “Simulating the Demo Model” on page 4-11

Overview of a Demo Model

This chapter illustrates how Simulink software can model a dynamic control system, using an example that simulates the thermodynamics of a house. The system models the outdoor environment, the thermal characteristics of a house, and the house heating system.

This chapter allows you to explore common Simulink modeling tasks, including:

- Grouping multiple blocks in a model into a single *subsystem*, simplifying the block diagram
- Customizing the appearance of blocks using the masking feature
- Simulating a model and observing the results using a Scope block
- Changing the input parameters of the model to investigate how the system responds
- Importing data from the MATLAB workspace into a model before simulation
- Exporting simulation data from the model back to the MATLAB workspace

Understanding the Demo Model

In this section...
“Opening the Demo Model” on page 4-3
“Anatomy of the Demo Model” on page 4-5
“Subsystems in the Demo Model” on page 4-6
“Masking Subsystems” on page 4-9

Opening the Demo Model

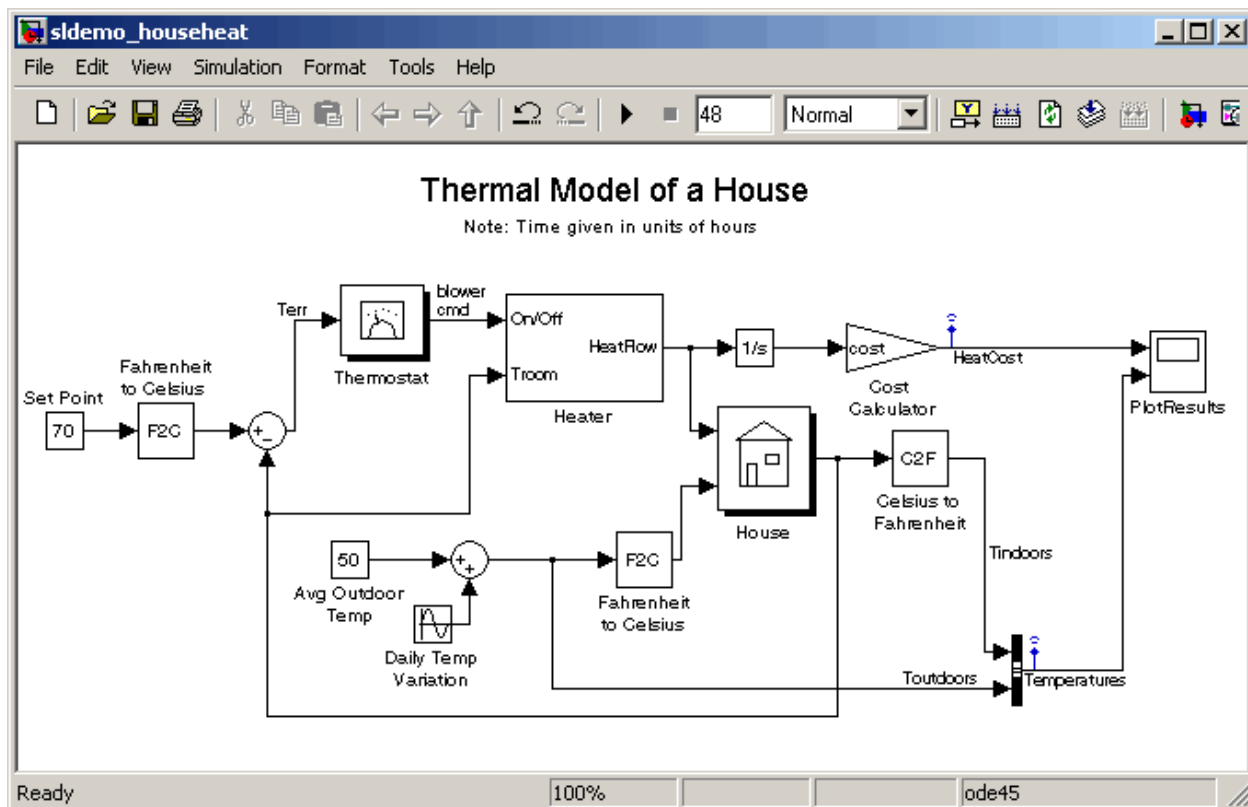
The demo model described in this chapter is called `sldemo_househeat`. It models the heating system and thermodynamics of a house.

- 1 Start MATLAB.
- 2 In the MATLAB Command Window, enter

```
sldemo_househeat
```

The Simulink editor opens with the demo model.

4 Modeling a Dynamic Control System



Anatomy of the Demo Model

The demo model defines the dynamics of the outdoor environment, the thermal characteristics of the house, and the house heating system. It allows you to simulate how the thermostat setting and outdoor environment affect the indoor temperature and cumulative heating costs.

The demo model includes many of the same blocks you used to create the simple model in Chapter 3, “Creating a Simulink Model”. These include:

- A *Scope block* (labeled `PlotResults`) on the far right, displays the simulation results.
- A *Mux block* at the bottom right, combines the indoor and outdoor temperature signals for the Scope.
- A *Sine Wave block* (labeled `Daily Temp Variation`) at the bottom left, provides one of three data sources for the model.

In the demo model, the thermostat is set to 70 degrees Fahrenheit. The system models fluctuations in outdoor temperature by applying a sine wave with amplitude of 15 degrees to a base temperature of 50 degrees.

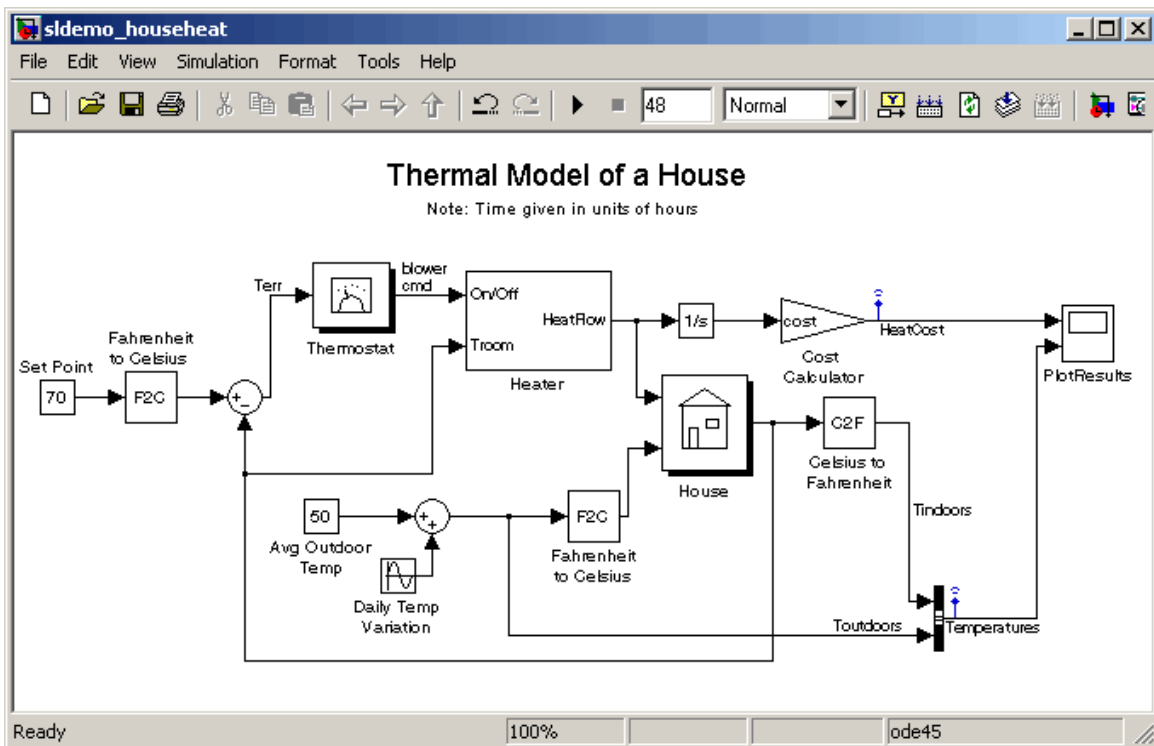
The three data inputs (sources) are provided by two Constant blocks (labeled `Set Point` and `Avg Outdoor Temp`), and the Sine Wave block (labeled `Daily Temp Variation`). The Scope block labeled `PlotResults` is the one output (sink).

Subsystems in the Demo Model

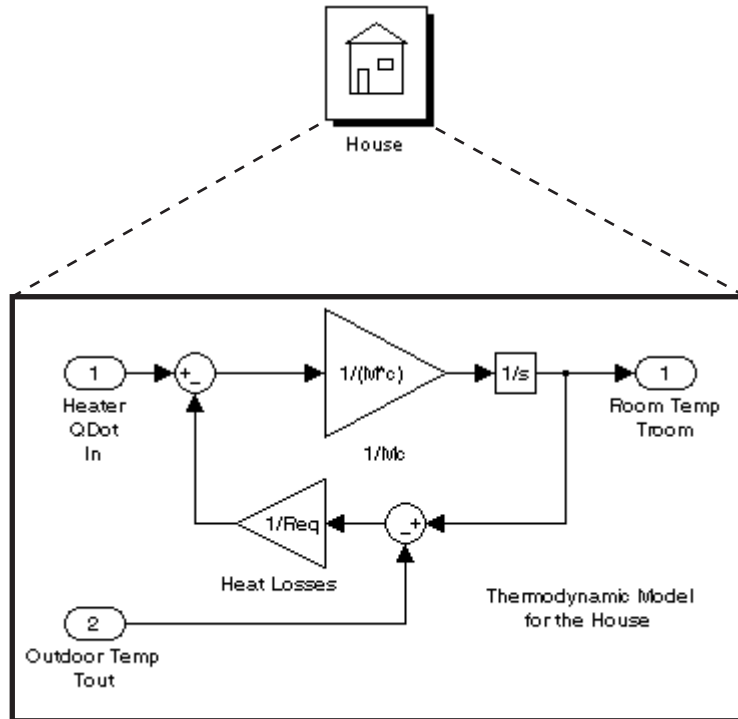
The `sldemo_househeat` demo model uses *subsystems* to simplify the appearance of the block diagram, create reusable components, and customize the appearance of blocks. A subsystem is a hierarchical grouping of blocks encapsulated by a single Subsystem block.

The demo model uses the following subsystems:

- Thermostat
- Heater
- House
- Fahrenheit to Celsius
- Celsius to Fahrenheit



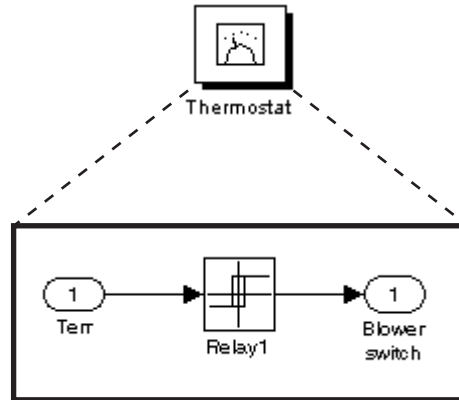
Subsystems can be complex and contain many blocks that might otherwise clutter a diagram. For example, double-click the House subsystem to open it.



Contents of House subsystem

You can see that the subsystem receives heat flow and external temperature as inputs, which it uses to compute the current room temperature. You could leave each of these blocks in the main model window, but combining them as a subsystem helps simplify the block diagram.

Subsystems can also be simple and contain only a few blocks. For example, double-click the Thermostat subsystem to open it.



Contents of Thermostat subsystem

This subsystem models the operation of a thermostat, determining when the heating system is on or off. It contains only one Relay block, but logically represents the thermostat in the block diagram.

Subsystems are also reusable, enabling you to implement an algorithm once and use it multiple times. For example, the model contains two instances of identical subsystems named *Fahrenheit to Celsius*. These subsystems convert the inside and outside temperatures from degrees Fahrenheit to degrees Celsius.

Creating a Subsystem

Creating a subsystem allows you to group related blocks into one subsystem block.

- 1 In the demo model window, select the Thermostat block.
- 2 Press the Shift key, and then select the Heater block.
- 3 Select **Edit > Create Subsystem**.

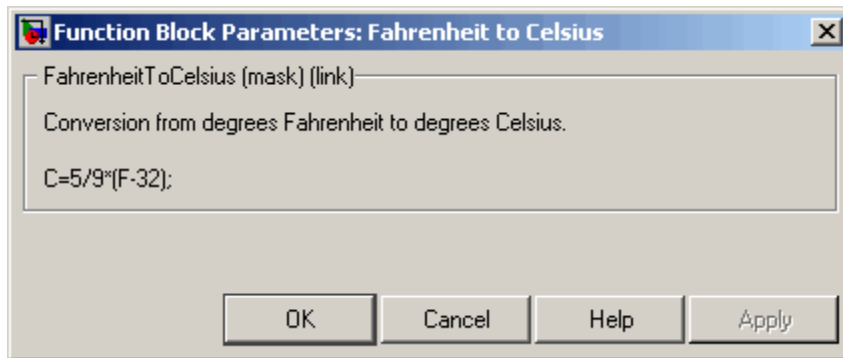
The software creates a Subsystem block containing the Thermostat and Heater blocks.

- 4 To return the model to its original configuration, select **Edit > Undo Create Subsystem**.

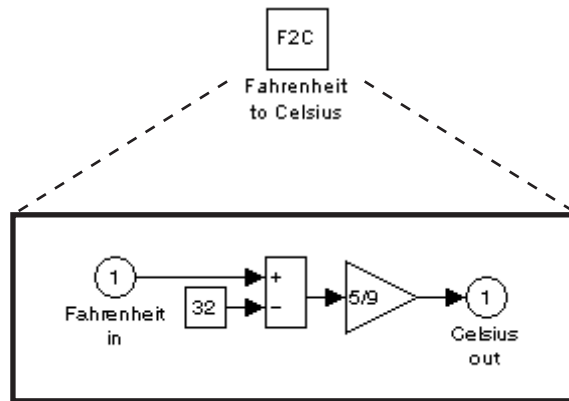
For more information about working with subsystems, see “Creating Subsystems” in the Simulink User’s Guide.

Masking Subsystems

You can customize the appearance of a subsystem by using a process known as *masking*. Masking a subsystem allows you to specify a unique icon and dialog box for the Subsystem block. For example, the House and Thermostat subsystems display custom icons that depict physical objects, while the conversion subsystems display custom dialog boxes when you double-click them.



To view the underlying blocks in the conversion subsystem, right-click the subsystem block, then select **Look Under Mask**.



Contents of Fahrenheit to Celsius subsystem

Creating a Subsystem Mask

To mask a subsystem:

- 1 In the demo model window, select the Heater block.
- 2 Select **Edit > Mask Subsystem**.
The Mask Editor dialog box opens.
- 3 From the **Command** list, select `disp` (show text in center of block) to view the syntax of this command.
- 4 In the **Icon Drawing commands** field, enter `disp('HEATER')`.
- 5 Click **OK**.

The software masks the subsystem block with the text you entered.

For more information about masking subsystems, see “Working with Block Masks” in the Simulink User’s Guide.

Simulating the Demo Model

In this section...

“Running the Simulation” on page 4-11

“Modifying Simulation Parameters” on page 4-13

“Importing Data from the MATLAB Workspace” on page 4-20

“Exporting Simulation Data to the MATLAB Workspace” on page 4-25

Running the Simulation

Simulating the model allows you to observe how the thermostat setting and outdoor environment affect the indoor temperature and the cumulative heating cost.

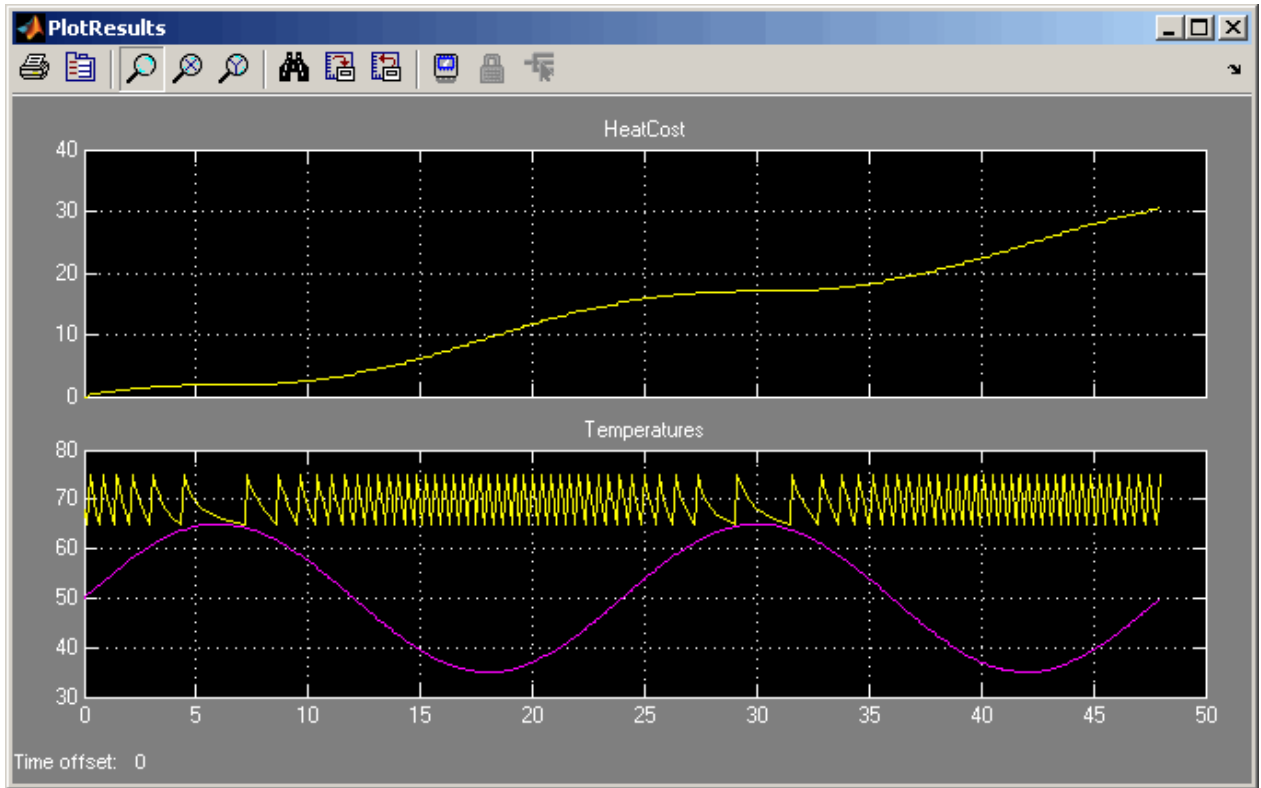
- 1** In the demo model window, double-click the Scope block named PlotResults.

The software opens a Scope window that contains two axes with the labels HeatCost and Temperatures.

- 2** Select **Simulation > Start**.

The software simulates the model. As the simulation runs, the cumulative heating cost appears on the HeatCost graph at the top of the Scope window. The indoor and outdoor temperatures appear on the Temperatures graph as yellow (top) and magenta (bottom) signals, respectively.

4 Modeling a Dynamic Control System



Modifying Simulation Parameters

One of the most powerful benefits of modeling a system with Simulink is the ability to interactively define the system inputs and observe changes in the behavior of your model. This allows you to quickly evaluate your model and validate the simulation results. This section describes:

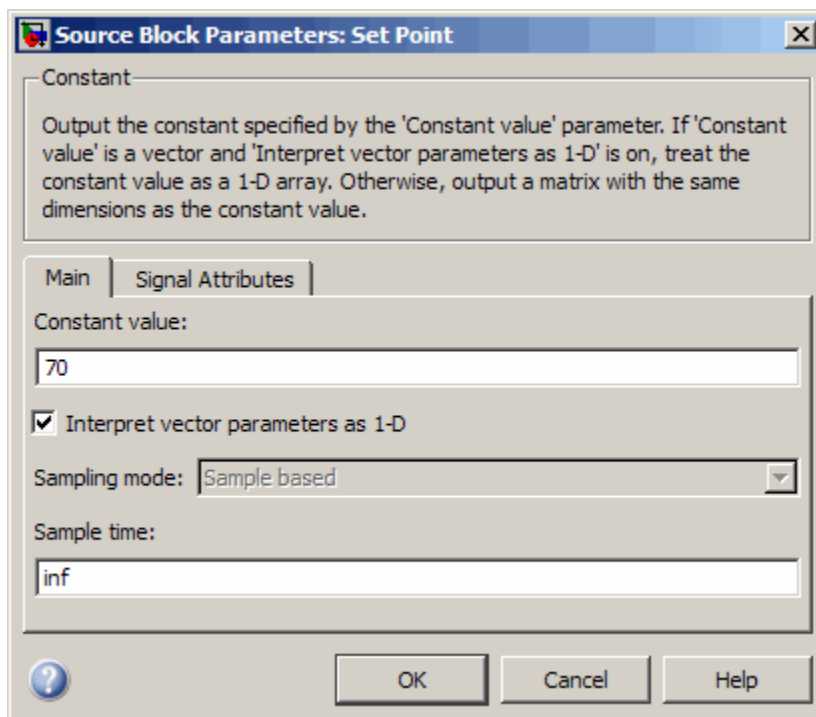
- “Changing the Thermostat Setting” on page 4-13
- “Changing the Average Outdoor Temperature” on page 4-16
- “Changing the Daily Temperature Variation” on page 4-18

Changing the Thermostat Setting

Change the thermostat setting to 68 degrees Fahrenheit and observe how the model responds.

- 1 In the demo model window, double-click the Set Point block.

The Source Block Parameters dialog box opens.



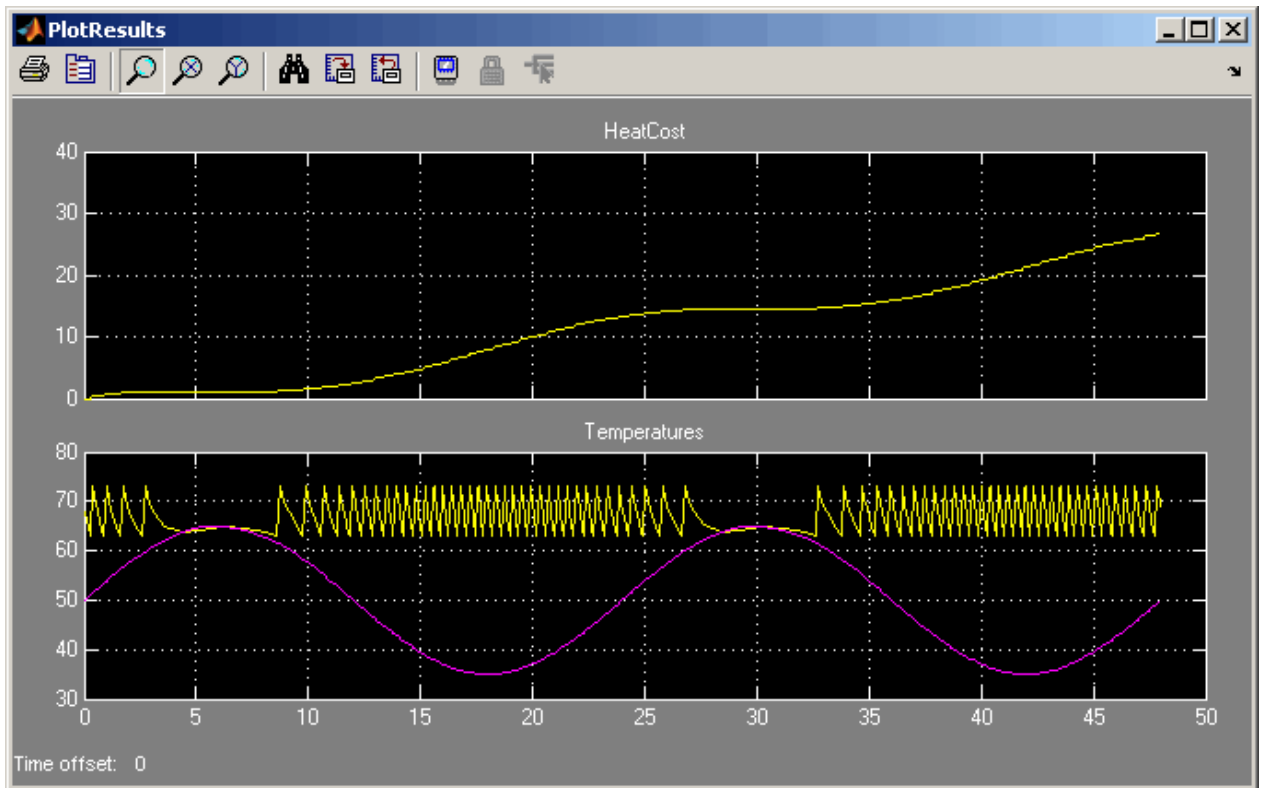
2 In the **Constant value** field, enter 68.

3 Click **OK**.

The software applies your changes.

4 To rerun the simulation, select **Simulation > Start**.

The software simulates the model. In the Scope window, notice that a lower thermostat setting reduces the cumulative heating cost.

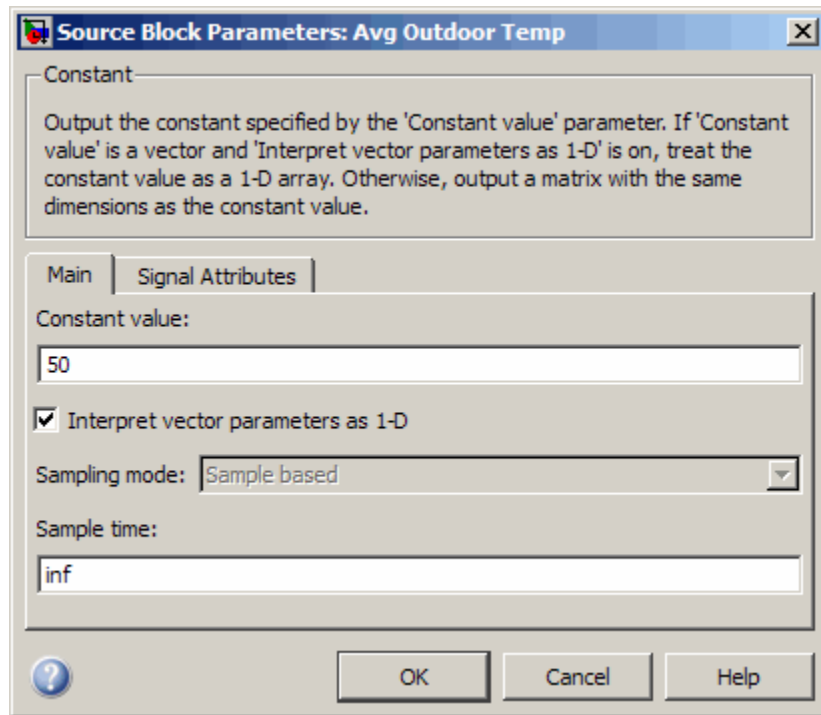


Changing the Average Outdoor Temperature

Change the average outdoor temperature to 45 degrees Fahrenheit and observe how the model responds.

- 1 In the demo model window, double-click the Avg Outdoor Temp block.

The Source Block Parameters dialog box opens.



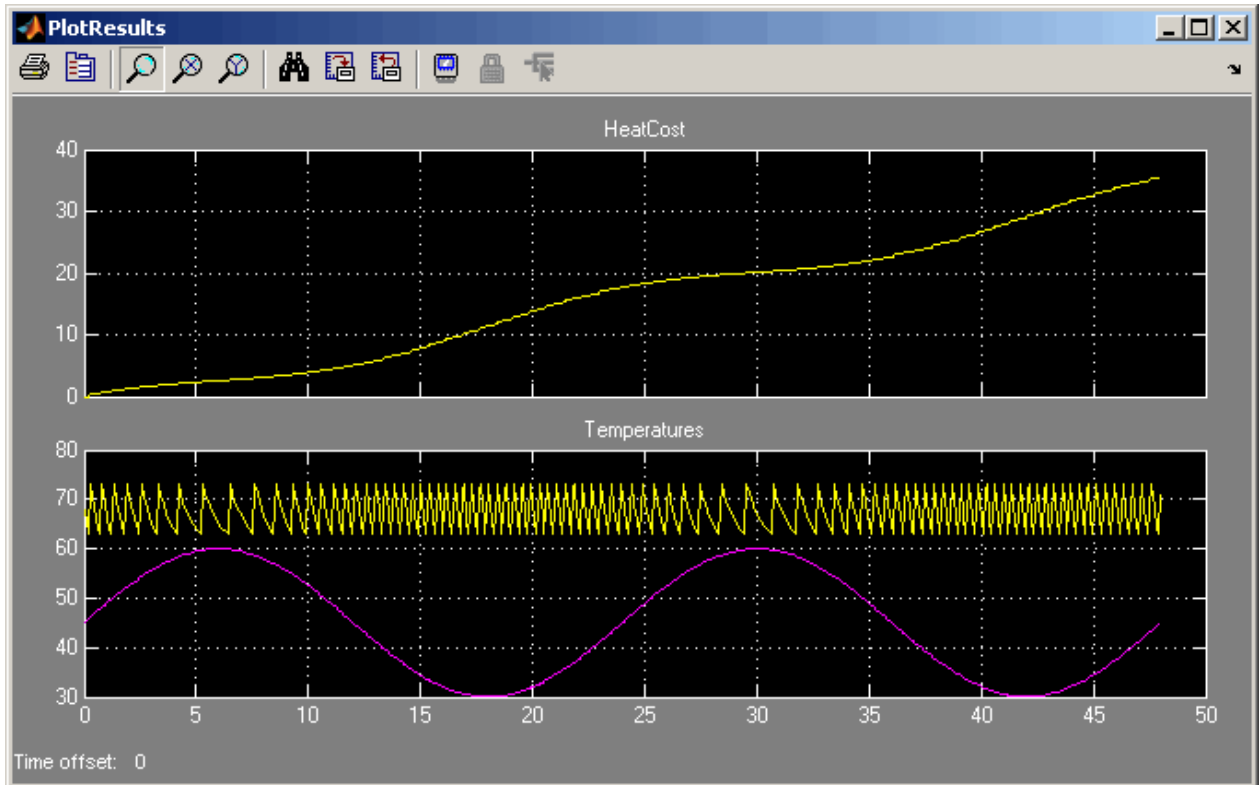
- 2 In the **Constant value** field, enter 45.

- 3 Click **OK**.

The software applies your changes and closes the dialog box.

- 4 To rerun the simulation, select **Simulation > Start**.

The software simulates the model dynamics. In the Scope window, notice that a colder outdoor temperature increases the cumulative heating cost.

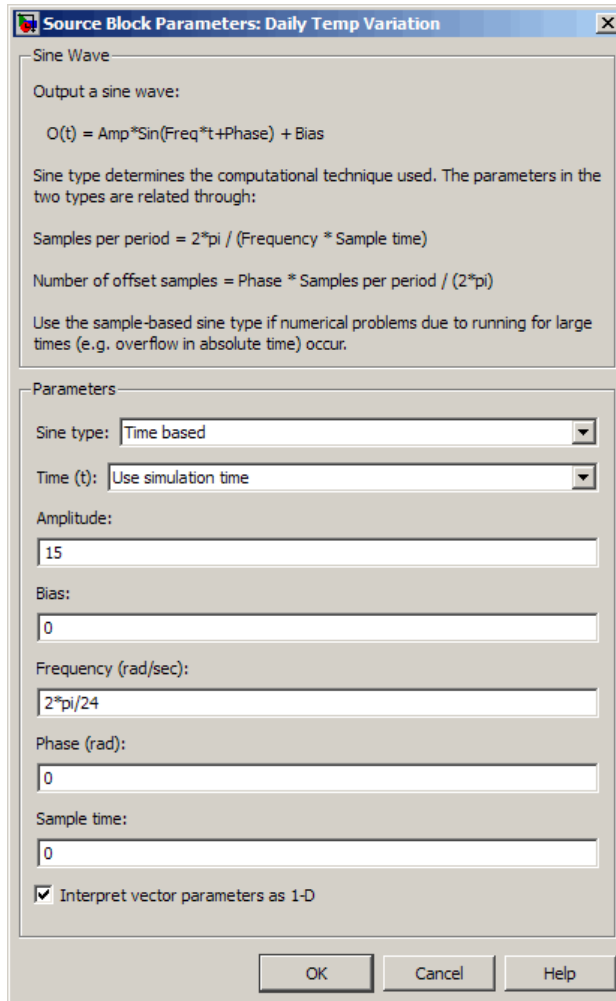


Changing the Daily Temperature Variation

Decrease the temperature variation to see how the model responds.

- 1 In the demo model window, double-click the Daily Temp Variation block.

The Source Block Parameters dialog box opens.



The dialog box is titled "Source Block Parameters: Daily Temp Variation". It contains the following sections and fields:

- Sine Wave**
 - Output a sine wave:
 - Equation: $O(t) = \text{Amp} * \sin(\text{Freq} * t + \text{Phase}) + \text{Bias}$
 - Text: Sine type determines the computational technique used. The parameters in the two types are related through:
 - Equation: $\text{Samples per period} = 2 * \pi / (\text{Frequency} * \text{Sample time})$
 - Equation: $\text{Number of offset samples} = \text{Phase} * \text{Samples per period} / (2 * \pi)$
 - Text: Use the sample-based sine type if numerical problems due to running for large times (e.g. overflow in absolute time) occur.
- Parameters**
 - Sine type: Time based (dropdown)
 - Time (t): Use simulation time (dropdown)
 - Amplitude: 15 (text field)
 - Bias: 0 (text field)
 - Frequency (rad/sec): $2 * \pi / 24$ (text field)
 - Phase (rad): 0 (text field)
 - Sample time: 0 (text field)
 - Interpret vector parameters as 1-D

Buttons: OK, Cancel, Help

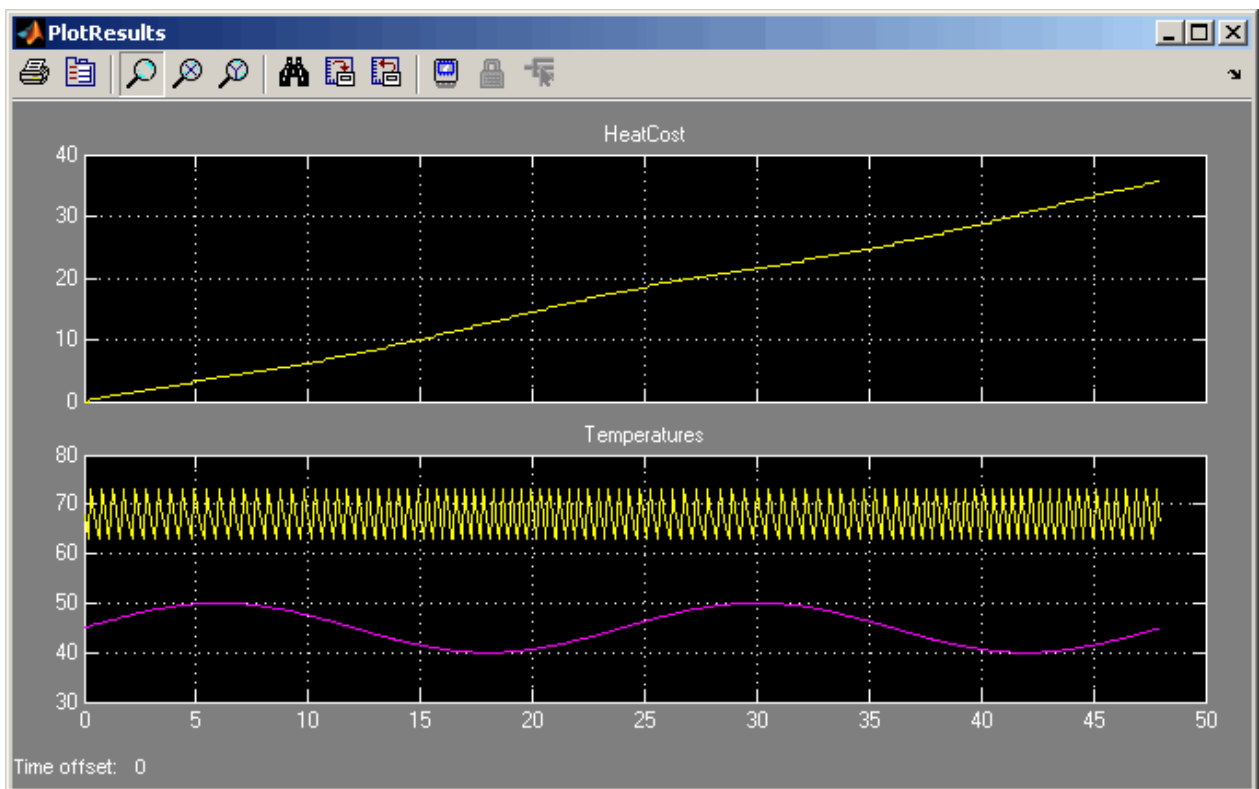
- 2 In the **Amplitude** field, enter 5.

3 Click **OK**.

The software applies your changes and closes the dialog box.

4 To rerun the simulation, select **Simulation > Start**.

The software simulates the model. In the Scope window, notice that a more stable outdoor temperature alters the frequency with which the heater operates.



Importing Data from the MATLAB Workspace

Simulink also allows you to import data from the MATLAB workspace to the model input ports. This allows you to import actual physical data into your model. For information about other data import capabilities, see “Importing and Exporting Simulation Data” in the Simulink User’s Guide.

Note In this example, you will create a vector of temperature data in MATLAB, and use that data as an input to the Simulink model.

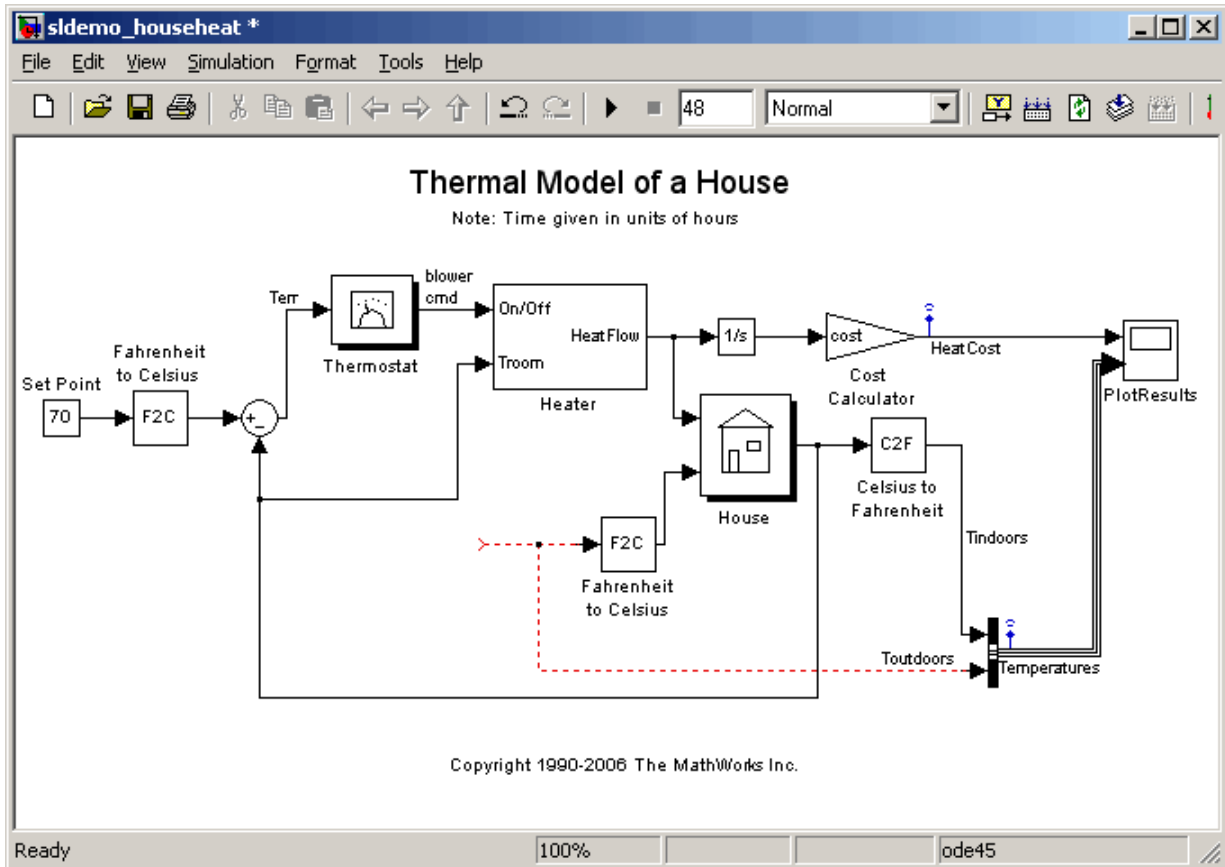
To import data from the MATLAB workspace:

- 1 In the MATLAB Command Window, create time and temperature data by entering the following commands:

```
x = (0:0.01:4*pi)';  
y = 32 + (5*sin(x));  
z = linspace(0,48,1257)';
```

- 2 In the Simulink model window, select the Avg Outdoor Temp block, and then press the **Delete** key to delete it.
- 3 Delete the following items from the model in the same way:
 - Daily Temp Variation block
 - Two input signal lines to the Sum block
 - Sum block

The model should now look similar to the following figure. Notice that the output signal from the Sum block changes to a red, dotted line, indicating that it is not connected to a block.



4 In the demo model window, select **View > Library Browser**.

The Simulink Library Browser window opens.

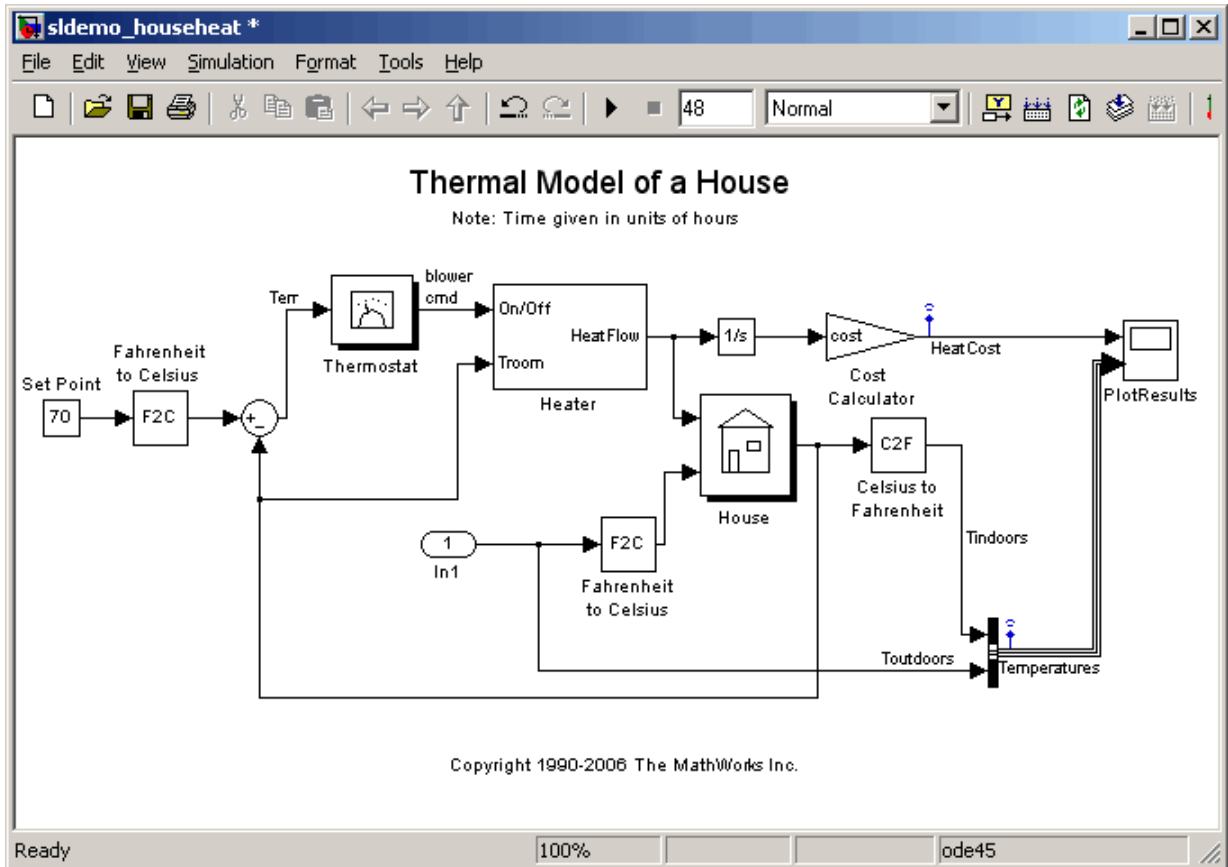
5 In the Simulink Library Browser, select the **Sources** library.

6 From the **Sources** library right pane, select the **In1** block, and then drag it to the model window.

An In1 block appears in the model window.

7 Connect the dotted line (originally connected to the Sum block) to the In1 block.

4 Modeling a Dynamic Control System

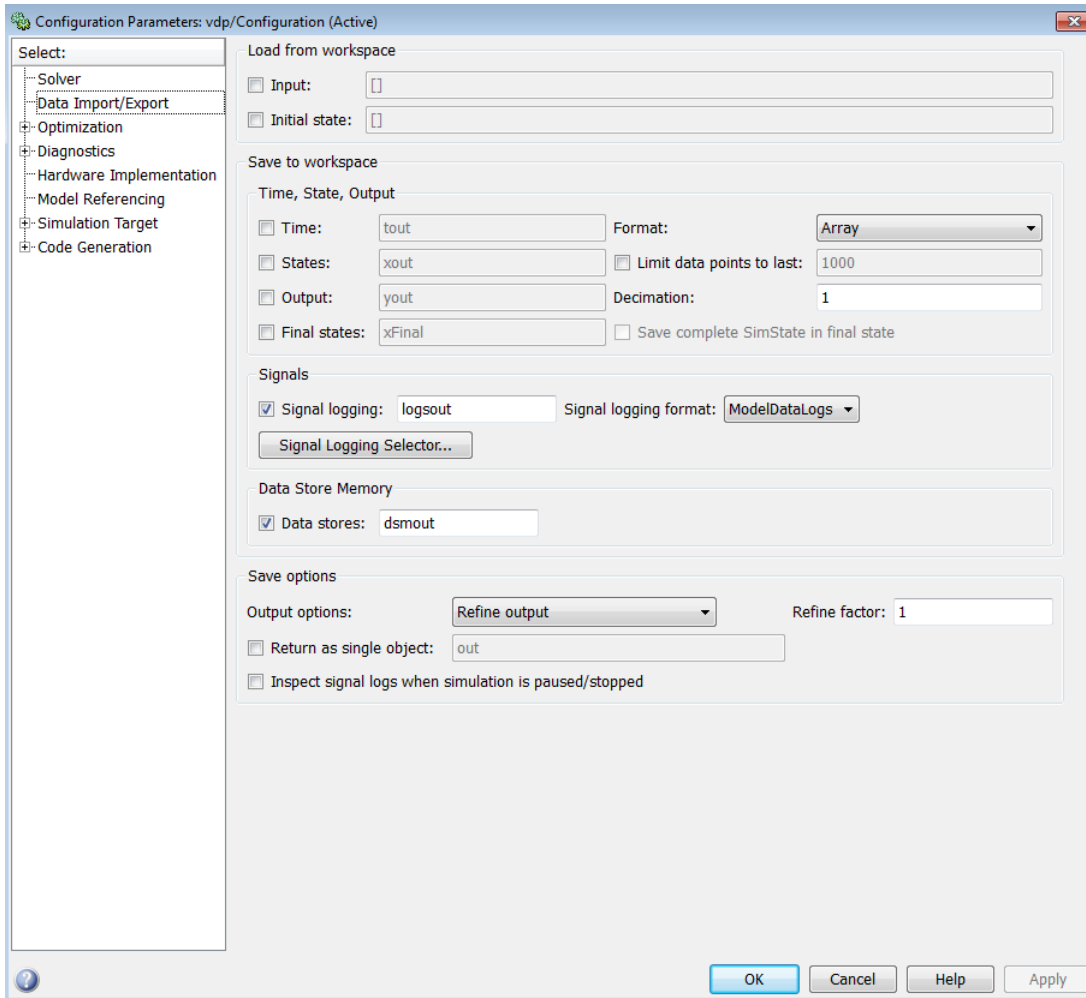


8 In the model window, select **Simulation > Configuration Parameters**.

The Configuration Parameters dialog box appears.

9 In the menu on the left side of the dialog, select **Data Import/Export**.

The **Data Import/Export** pane opens.



10 Select the **Input** check box.

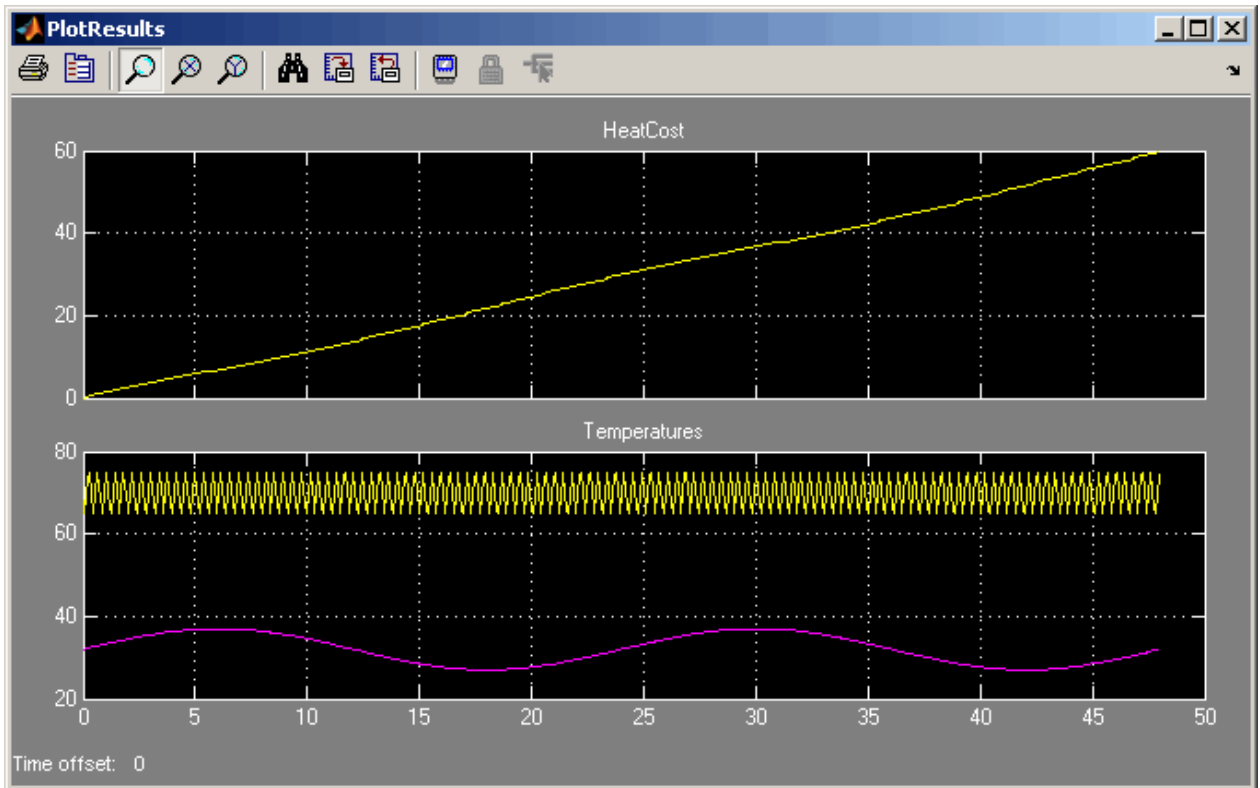
11 In the **Input** field, enter [z , y].

12 Click **OK**.

The software applies your changes and closes the dialog box.

13 To rerun the simulation, select **Simulation > Start**.

The software simulates the model. In the Scope window, notice that the model ran using the imported data, showing colder temperatures and higher heat use.



Exporting Simulation Data to the MATLAB Workspace

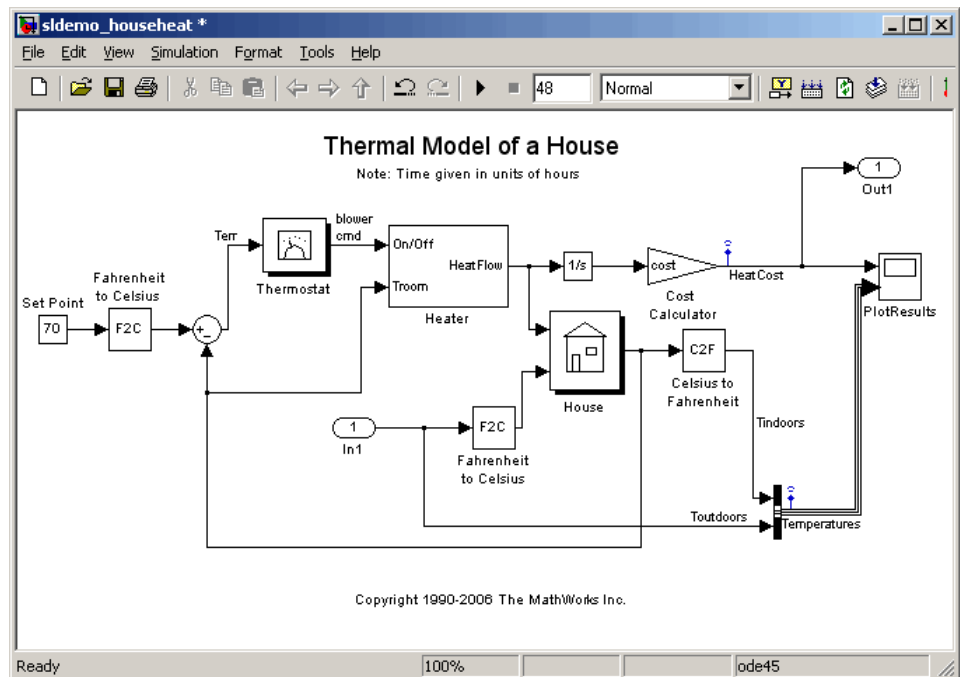
Once you have completed a model, you may want to export your simulation results to MATLAB workspace for further data analysis or visualization. For information about additional data export capabilities, see “Exporting Simulation Data”.

To export the HeatCost data from the model to the MATLAB workspace:

- 1 In the Simulink Library Browser window, select the **Sinks** library.
- 2 From the **Sinks** library, select the **Out1** block, and then drag it to the top right of the demo model window.

An Out1 block appears in the model window.

- 3 Draw a *branch line* from the HeatCost signal line to the Out1 block. For more information, see “Drawing a Branch Line” on page 3-9.

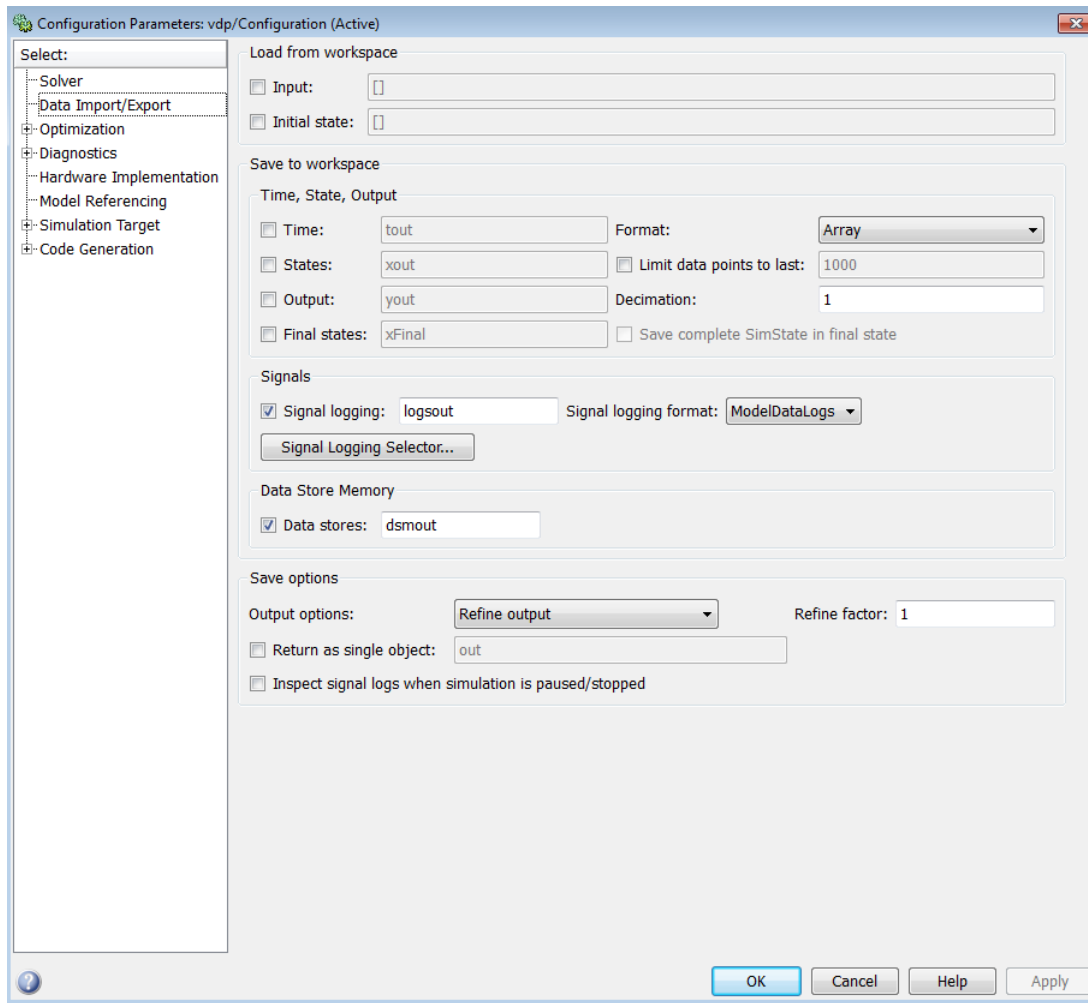


4 Select **Simulation > Configuration Parameters**.

The Configuration Parameters dialog box opens.

5 From the menu on the left side of the dialog box, select **Data Import/Export**.

The **Data Import/Export** pane opens.



6 Select the **Time** and **Output** check boxes.

7 Click **OK**.

The software applies your changes and closes the dialog box.

8 To rerun the simulation, select **Simulation > Start**.

The software simulates the model and saves the time and HeatCost data to the MATLAB workspace. Notice that the tout and yout variables now appear in the MATLAB workspace.

Completing a Basic Simulation Workflow

- “Overview” on page 5-2
- “Importing Data from a Microsoft® Excel Spreadsheet” on page 5-4
- “Configuring the Model” on page 5-9
- “Simulating the Model” on page 5-10
- “Comparing Signals in the Simulation Data Inspector” on page 5-13
- “Creating, Saving, and Printing a Figure in Simulation Data Inspector” on page 5-15

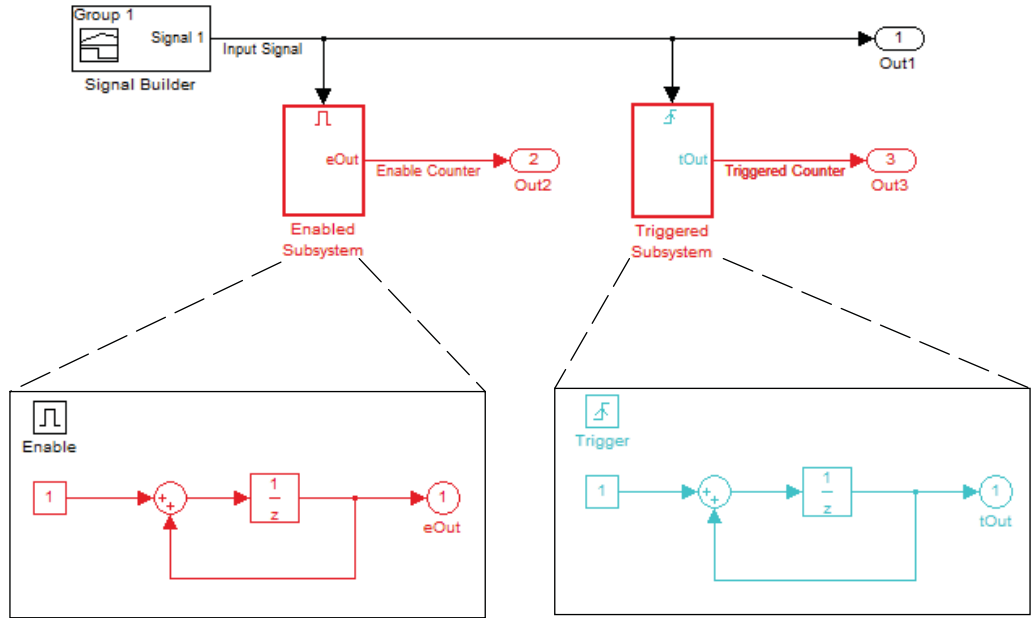
Overview

Simulink provides two important features that allow you to complete a basic simulation workflow: the Signal Builder block and the Simulation Data Inspector. The Signal Builder block allows you to import an input signal to your model. The Simulation Data Inspector allows you to efficiently view model output signals and to compare them.

This guide uses a simple model, `sldemo_2counters.mdl`, to demonstrate the following typical workflow:

- 1** Import signal data from a Microsoft® Excel® spreadsheet into a Signal Builder block.
- 2** Configure for the logging of output signals during simulation.
- 3** Simulate the model.
- 4** View and compare the output signals in the Simulation Data Inspector.
- 5** Create, print, and save a plot from the Simulation Data Inspector.

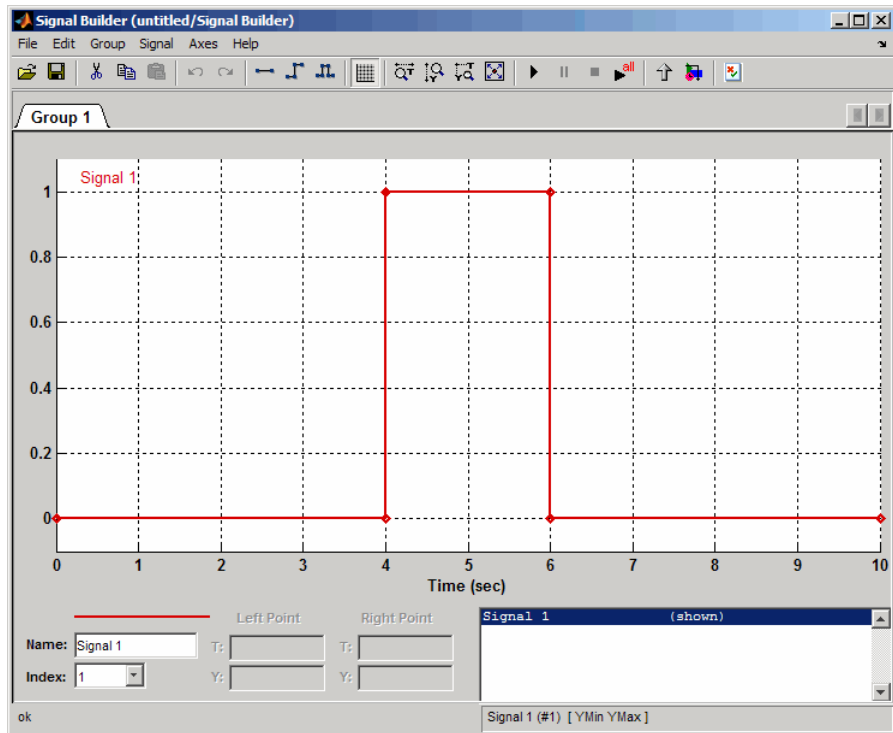
The `sldemo_2counters` model consists of a Signal Builder block and two subsystems—an Enabled Subsystem and a Triggered Subsystem—each of which serves as a counter circuit. For the Enabled Subsystem, the counter increments when the subsystem is enabled and subsequently retains its output while the subsystem is disabled. Because the Enable port is set to reset states when enabling, the counter returns to zero when the subsystem is re-enabled. In contrast, the counter inside the Triggered Subsystem increments for each rising edge of the control signal.



Importing Data from a Microsoft Excel Spreadsheet

You can use a Signal Builder block as the input source for your model and to import your own signal data to the model.

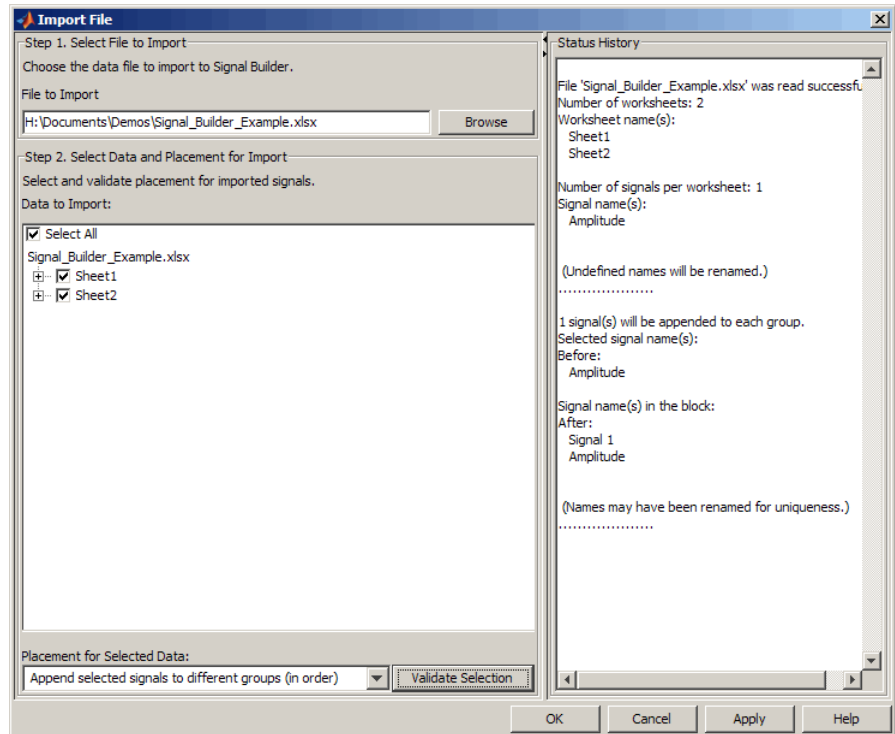
- 1 Start the MATLAB software. If you need instructions, consult your MATLAB documentation.
- 2 At the MATLAB command line, enter `sldemo_2counters`. This command starts the Simulink software and opens the model.
- 3 In the model window, double-click the Signal Builder block. The Signal Builder window opens and displays one data group with a default signal.



- 4 Prepare for importing a Microsoft Excel spreadsheet that contains data on two sheets by creating a second group. Each sheet of data requires its own

group. From the menu bar, select **Group > Copy**. A second tab, **Group 2**, appears.

- 5** Import the data by selecting **File > Import from File**. An Import File dialog box opens.
- 6** Use the **Browse** button to open the `sldemo_signal_builder.xls` file.
- 7** Under **Data to Import** , select the **Select All** check box.
- 8** Under **Placement for Selected Data**, select **Append selected signals to different groups**. This option causes the Signal Builder to convert the data on each sheet into a group and to enable the **Validate Selection** button.
- 9** Click the **Validate Selection** button.
- 10** Review the **Status History** pane.



The history indicates the following information:

- The data importation is successful.
- The Signal Builder block recognizes one signal per sheet.
- The Signal Builder block recognizes the signal name, Amplitude, from the Excel® spreadsheet.

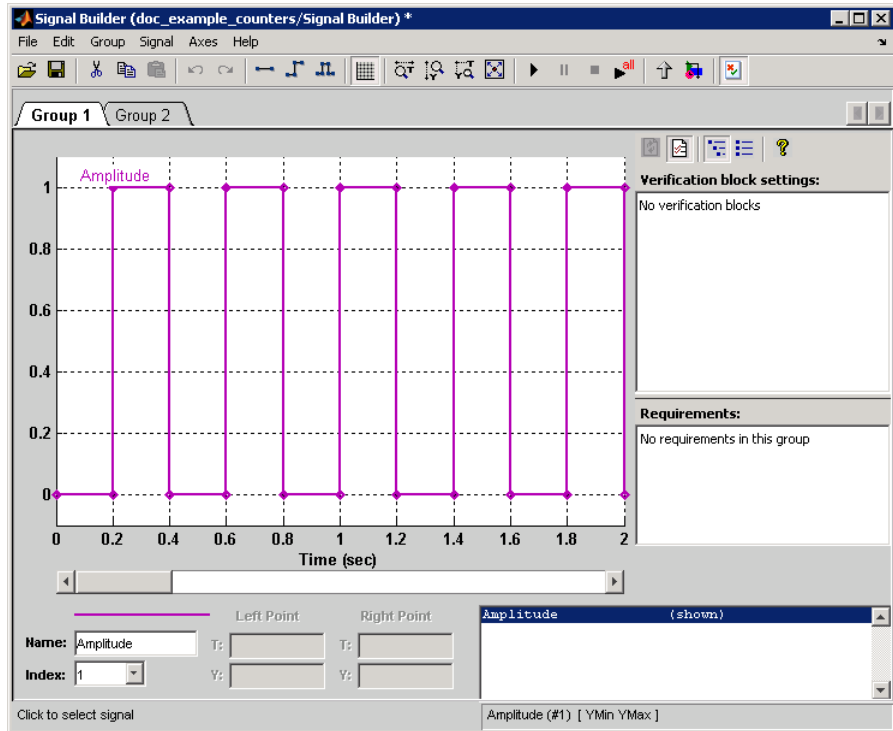
11 Click **OK**. The window closes.

12 In the Signal Builder window, delete the red default signal from Group 1 by clicking it and then pressing the **Delete** button on your keyboard. The signal is automatically deleted from Group 2 as well.

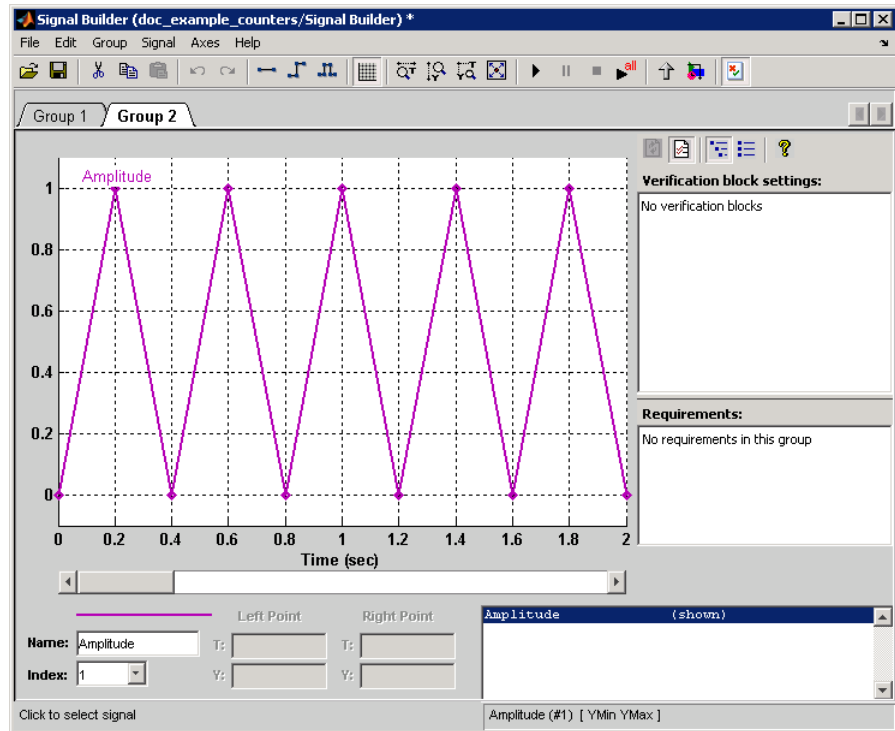
13 From the Signal Builder menu bar, select **Axes > Set T Display Limits**. A dialog box opens. Set the maximum time limit to 2.0.

- 14 Save your Signal Builder data to the model. In the Signal Builder window, select **File > Save**.

Your signal groups appear as shown. Keep this window open.



5 Completing a Basic Simulation Workflow



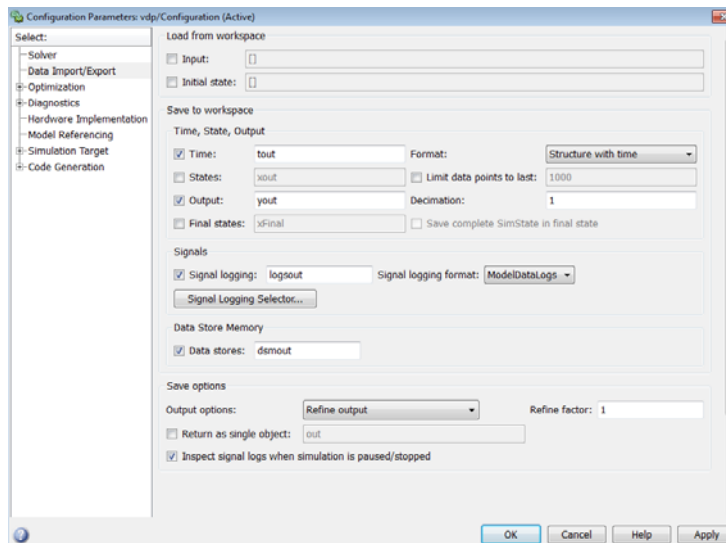
You can now set up the model to log output signals. See “Configuring the Model” on page 5-9.

Configuring the Model

For Logging Output Signals

To log and display your output, you must configure the data import and export parameters of the model.

- 1 In the model window, select **Simulation > Configuration Parameters**. The Configuration Parameters dialog box opens to the **Solver** pane.
- 2 Navigate to the **Data Import/Export** pane by clicking its title in the tree.
- 3 Select the following check boxes:
 - **Time**
 - **Output**
 - **Signal logging**
 - **Inspect signal logs when simulation is paused/stopped**
- 4 Change **Format** from Array to Structure with time.



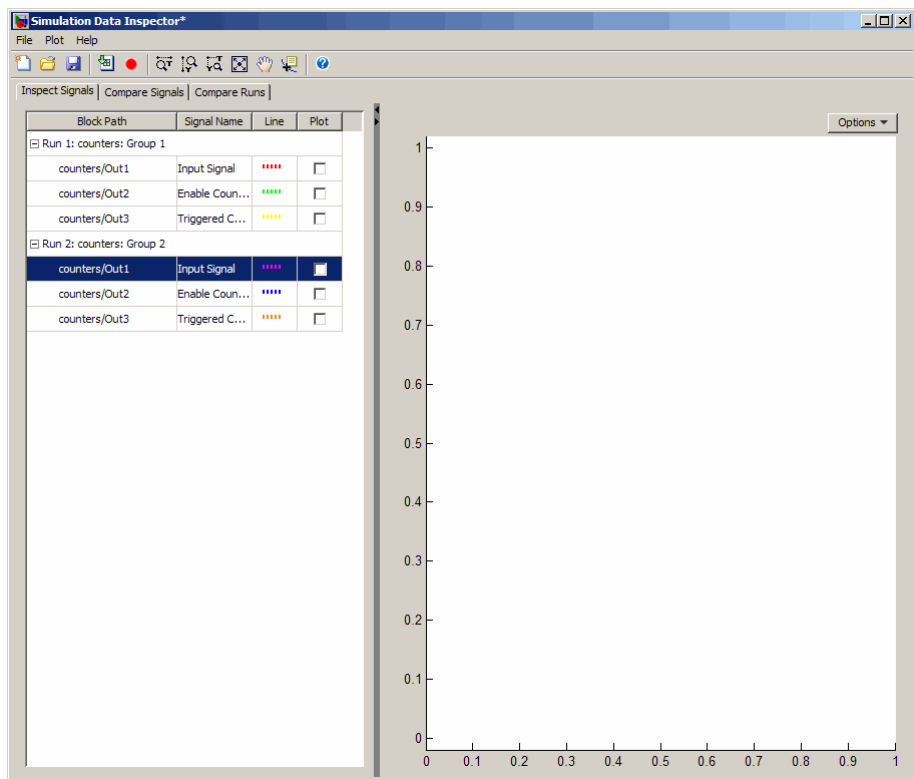
Simulating the Model

With a single click, you can run this model for an input and then view and compare the results in the Simulation Data Inspector.

- 1 In the Signal Builder toolbar, click the **Run all** button.

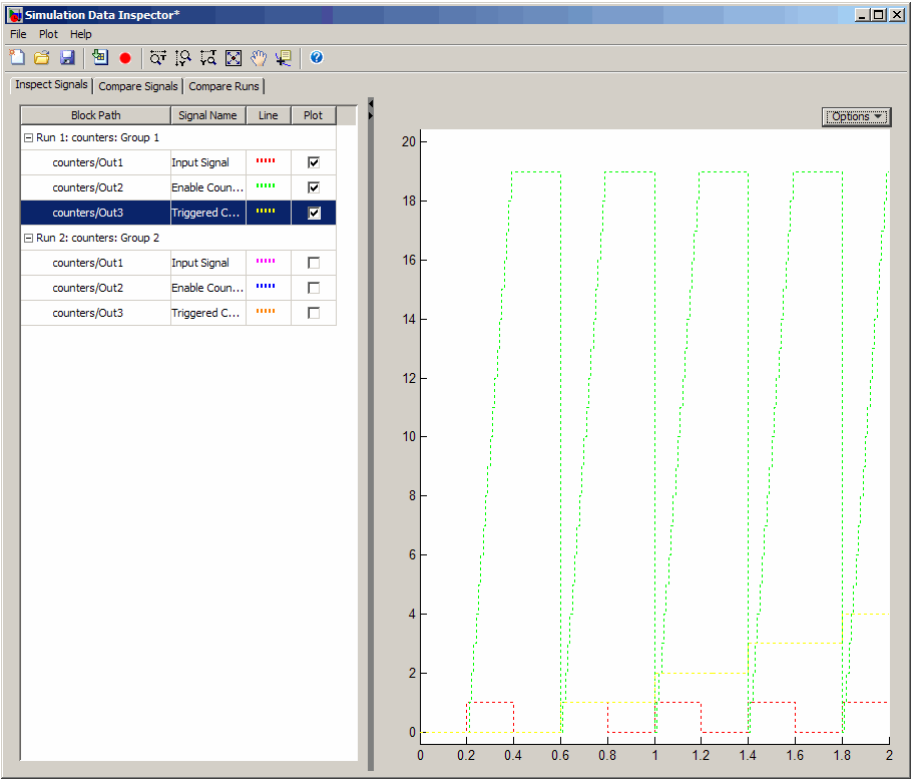


Simulink runs the model from 0.0 s to 2.00 s for the Group 1 and Group 2 amplitude signals. And because you selected **Inspect signal logs when simulation is paused/stopped**, the Simulation Data Inspector automatically opens and contains your output data in Run 1 and Run 2.

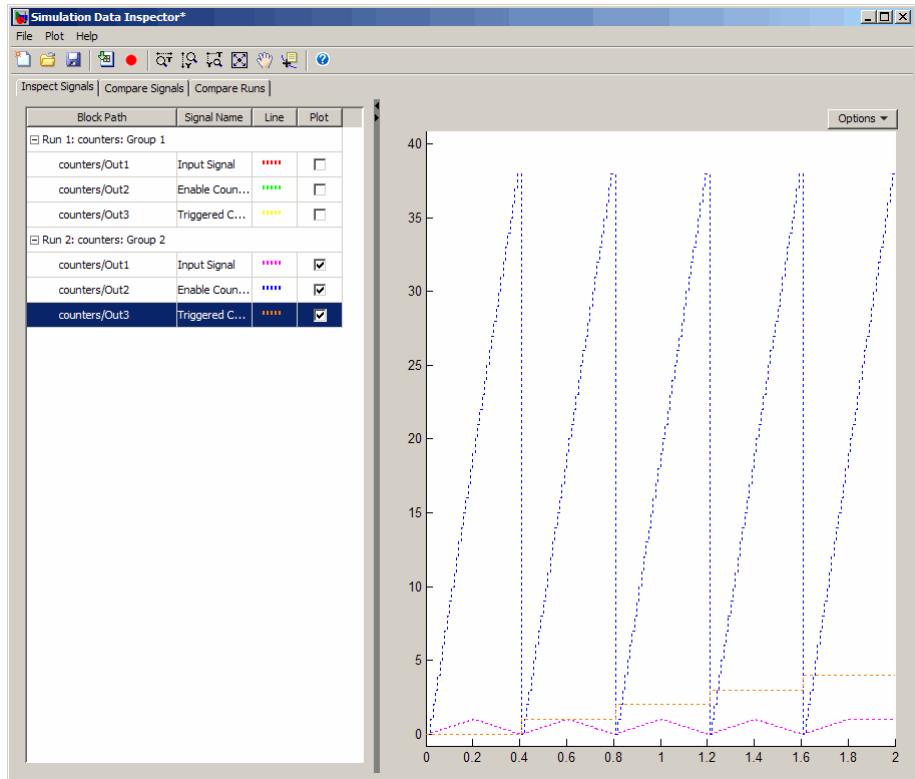


2 Select the signals that you want to display.

If you select all output signals for Run 1, three line plots appear in the right pane of the Simulation Data Inspector window.



3 If you select the output signals for only Run 2, the plotted lines appear as follows.



Now, you can inspect the results further. See “Comparing Signals in the Simulation Data Inspector” on page 5-13.

Comparing Signals in the Simulation Data Inspector

The Simulation Data Inspector allows you to quickly compare two signals from the same or different runs.

1 In the Simulation Data Inspector window, click the **Compare Signals** tab. By default, five columns are displayed, including **Sig 1** and **Sig 2**.

2 In the **Sig 1** column, for Run 1, select the Enable Counter signal.

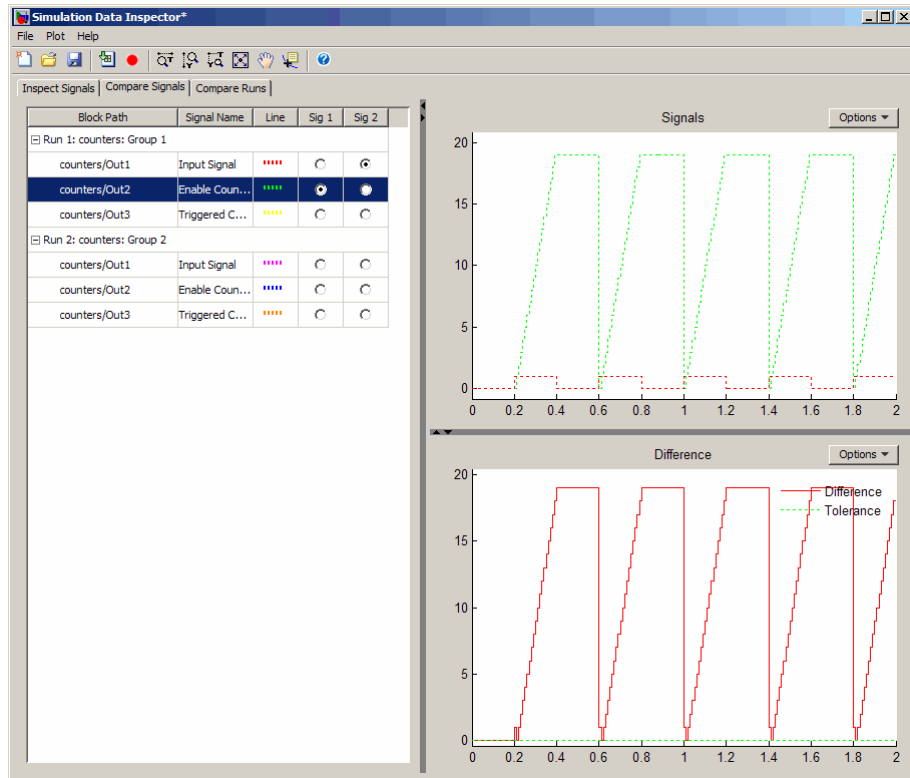
Selecting this signal as Sig 1 is important. The Inspector sets the maximum value for the *y*-axis of the plot based on this signal. Because the Enable Counter signal requires a maximum value of 20, and your next selection, the Control signal, requires a lesser value of 4, the Enable Counter is the better choice as the reference signal for plotting.

3 In the **Sig 2** column, for Run 1, select the Control signal.

4 Inspect the signals. In the top pane, the plot contains the two individual signals. In the bottom pane, a single plot represents the difference of these two signals.

5 In each pane, from the **Options** menu, select **Plot as line** to obtain smooth plots. The stair step lines become smooth.

5 Completing a Basic Simulation Workflow



To work with your plots as figures, see “Creating, Saving, and Printing a Figure in Simulation Data Inspector” on page 5-15.

Creating, Saving, and Printing a Figure in Simulation Data Inspector

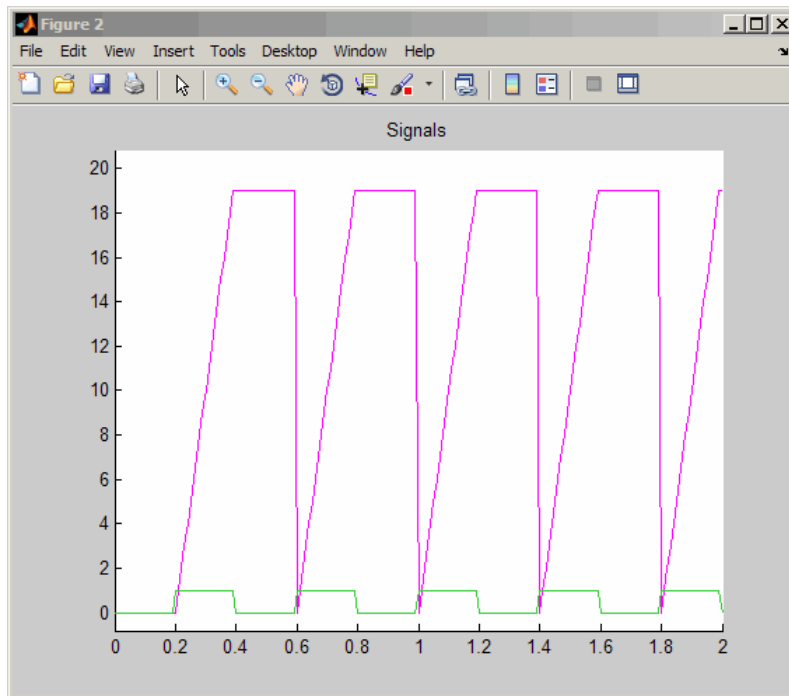
In this section...

“Creating a Figure from a Plot” on page 5-15

“Saving and Printing a Figure” on page 5-16

Creating a Figure from a Plot

- 1 From the **Options** menu in the Simulation Data Inspector, select **Plot data in new figure**. A new figure window opens with your plot.



Saving and Printing a Figure

- 1** To save your figure to your computer, in the Figures window, select **File > Save as**. A **Save as** dialog box opens.
- 2** Navigate to the folder where you want to save your figure.
- 3** In the **File name** field, enter the name of your figure.
- 4** From the **Save as type** menu, select a file format. If you keep the default, **MATLAB Figure (*.fig)**, then when you reopen the figure, the model also opens.
- 5** Click the **Save** button.
- 6** To print the figure, select **File > Print**.

A

adding blocks to a model 3-4

B

basics

 Simulink 2-1

block libraries

 descriptions 2-7

blocks

 connecting 3-7

 copying 3-4

 masking 4-9

 moving 3-6

 ports 3-7

 subsystems 4-6

browser

 library 2-6

C

closing

 model 3-14

Configuration Parameters dialog box 3-11

connecting

 blocks 3-7

 lines to input ports 3-7

control system

 model overview 4-5

 modeling 4-2

copying blocks to a model 3-4

creating

 new model 2-3 3-3

 subsystem 4-8

 subsystem mask 4-10

D

data

 exporting to MATLAB 4-25

 importing from MATLAB 4-20

demo model

 opening 4-3

 overview 4-5

 running 4-11

demos

 accessing 2-11

E

example model 3-2

exporting

 data from Simulink to MATLAB 4-25

H

help

 Simulink 2-10

house model

 overview 4-5

I

import data

 Excel spreadsheet 5-4

importing

 MATLAB data to Simulink 4-20

input data

 importing from MATLAB 4-20

input port 3-7

L

library

 descriptions 2-7

 searching 2-7

 selecting 2-7

 viewing 2-7

Library Browser

 opening 2-2

 overview 2-6

- using 3-4

- lines

- carrying the same signal 3-7

- connecting to input ports 3-7

M

- masking

- creating a mask 4-10

- overview 4-9

- MATLAB

- exporting data from Simulink 4-25

- importing data to Simulink 4-20

- Simulink interaction 1-4

- model configuration

- logging output 5-9

- model window

- moving blocks 3-6

- overview 2-9

- Model-Based Design

- overview 1-5

- process 1-6

- modeling

- control system 4-2

- process 1-6

- models

- closing 3-14

- control system example 4-5

- creating new 2-3

- creating simple 3-3

- empty 3-3

- opening 2-3

- opening existing 2-5

- simple example 3-2

- simulating 3-11

- modifying

- simulation parameters 4-13

- moving

- blocks in the model window 3-6

O

- online help

- accessing 2-10

- Simulink 2-10

- opening

- demo model 4-3

- existing model 2-5

- model 2-3

- Simulink 2-2

- output data

- exporting to MATLAB 4-25

- output port 3-7

- overview

- demo model 4-5

- Simulink 1-2

P

- parameters

- modifying 4-13

- product overview 1-2

- products

- related 1-9

R

- related products 1-9

- running

- demo model 4-11

- simulation 3-11

- Simulink 2-2

S

- sample

- model 3-2

- simulating models

- exercise 3-11

- simulation 5-10

- modifying parameters 4-13

- running 4-11
 - specifying options 3-11
 - starting 3-13
 - stopping 3-13
 - Simulink
 - basics 2-1
 - demos 2-11
 - getting help 2-10
 - Library Browser 2-6
 - MATLAB interaction 1-4
 - model window 2-9
 - online help 2-10
 - opening 2-2
 - overview 1-2
 - related products 1-9
 - starting 2-2
 - user interface 2-6
 - web resources 2-13
 - starting
 - MATLAB 2-2
 - simulation 3-13
 - Simulink 2-2
 - stopping
 - simulation 3-13
 - subsystems
 - creating 4-8
 - masking 4-9
 - overview 4-6
- T**
- thermo model
 - overview 4-5
- U**
- user interface
 - Simulink 2-6
- W**
- web resources
 - Simulink 2-13
 - workflow
 - overview 5-2