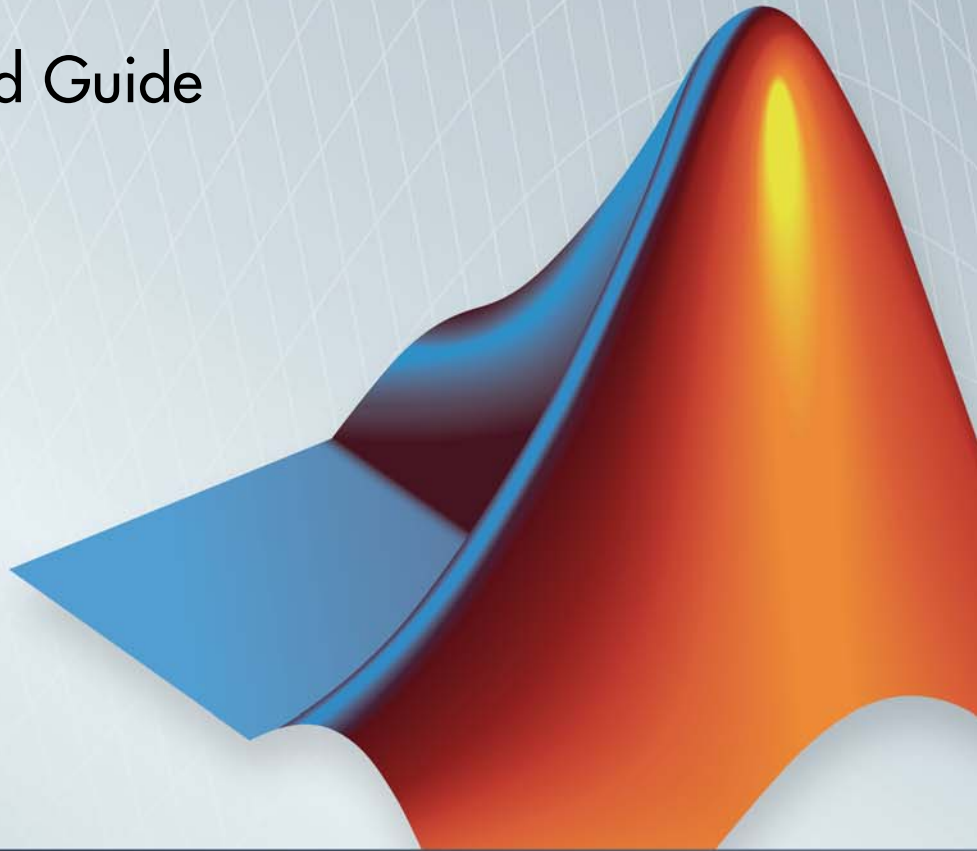


Simulink®

Getting Started Guide

R2013a



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 1990–2013 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)
September 2011	Sixth printing	Revised for Simulink 7.8 (Release 2011b)
March 2012	Seventh printing	Revised for Simulink 7.9 (Release 2012a)
September 2012	Eighth printing	Revised for Simulink 8.0 (Release 2012b)
March 2013	Ninth printing	Revised for Simulink 8.1 (Release 2013a)

Introduction

1

Product Description	1-2
Key Features	1-2
Modeling, Simulation, and Analysis with Simulink ...	1-3
Tool for Model-Based Design	1-3
Tool for Simulation	1-4
Tool for Analysis	1-4
Interaction with MATLAB Environment	1-4
Model-Based Design	1-5
What Is Model-Based Design?	1-5
Model-Based Design Process	1-6
Documentation and Additional Resources	1-9
Simulink Online Help	1-9
Simulink Examples	1-10
Web Site Resources	1-10

Simulink Software Basics

2

Start the Simulink Software	2-2
Open the Simulink Library Browser	2-2
Create a New Simulink Model	2-4
Open an Existing Model	2-5
Simulink User Interface	2-6
Simulink Library Browser	2-6
Simulink Editor	2-9

Create a Simulink Model

3

Create a Simple Model	3-2
Overview of the Simple Model	3-2
Create the Simple Model	3-3
Connect Blocks in the Simple Model	3-8
Simulate the Simple Model	3-13

Modeling a Dynamic Control System

4

Dynamic Control System Model	4-2
Anatomy of the Control System Model	4-3
Open the Example Model	4-3
Overview of the Example Model	4-4
Subsystems in the Example Model	4-6
Subsystems and Masks	4-10
Creating a Subsystem	4-11
Create a Subsystem Mask	4-12
Simulate the Control System Model	4-14
Run the Simulation	4-14
Change the Thermostat Setting	4-15
Change the Average Outdoor Temperature	4-17
Change the Daily Temperature Variation	4-19
Move Data Between MATLAB and Simulink Model ...	4-21
Import Data from the MATLAB Workspace	4-21
Export Simulation Data to the MATLAB Workspace	4-25

Simulation with Data Import and Signal Inspection .. 5-2

- Data Import and Logging Workflow 5-2
- Example Used in this Tutorial 5-2
- Import Signal Data 5-4
- Simulate Model 5-10
- Compare Signals 5-14
- Create, Save, and Print a Figure 5-16

Introduction

- “Product Description” on page 1-2
- “Modeling, Simulation, and Analysis with Simulink” on page 1-3
- “Model-Based Design” on page 1-5
- “Documentation and Additional Resources” on page 1-9

Product Description

Simulation and Model-Based Design

Simulink® is a block diagram environment for multidomain simulation and Model-Based Design. It supports system-level design, simulation, automatic code generation, and continuous test and verification of embedded systems. Simulink provides a graphical editor, customizable block libraries, and solvers for modeling and simulating dynamic systems. It is integrated with MATLAB®, enabling you to incorporate MATLAB algorithms into models and export simulation results to MATLAB for further analysis.

Key Features

- Graphical editor for building and managing hierarchical block diagrams
- Libraries of predefined blocks for modeling continuous-time and discrete-time systems
- Simulation engine with fixed-step and variable-step ODE solvers
- Scopes and data displays for viewing simulation results
- Project and data management tools for managing model files and data
- Model analysis tools for refining model architecture and increasing simulation speed
- MATLAB Function block for importing MATLAB algorithms into models
- Legacy Code Tool for importing C and C++ code into models

Modeling, Simulation, and Analysis with Simulink

In this section...

“Tool for Model-Based Design” on page 1-3

“Tool for Simulation” on page 1-4

“Tool for Analysis” on page 1-4

“Interaction with MATLAB Environment” on page 1-4

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 examples that model a wide variety of real-world phenomena (click the **Examples** link at the top of the Simulink Documentation Center top-level page).

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.

See “Model-Based Design” on page 1-5 for more information about the modeling process.

Tool for Simulation

After you define a model, you can simulate its dynamic behavior 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, you can use MATLAB scripts.

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

Interaction with 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 use 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.

Model-Based Design

In this section...

“What Is Model-Based Design?” on page 1-5

“Model-Based Design Process” on page 1-6

What Is 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 you continually refine 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

Model-Based Design 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 environment 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 `sldemo_househeat` example model of the heating system of a house is 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 a 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 system components, and you 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 that 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 “Create the 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 “Simulate the Simple Model” on page 3-13 for more information on running a simulation.

Validating the Simulation Results

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

Documentation and Additional Resources

In this section...
“Simulink Online Help” on page 1-9
“Simulink Examples” on page 1-10
“Web Site Resources” on page 1-10

Simulink Online Help

Simulink software provides comprehensive online help describing features, blocks, and functions with detailed procedures for common tasks.

Access online help from **Help** menus and context-sensitive block labels.

- From the Simulink Library Browser, select **Help > Simulink Help**.
- From the Simulink Editor, select **Help > Simulink > Simulink Help**.
- Right-click a Simulink block, and then select **Help**.
- From the model Configuration Parameters or a block parameters dialog box, right-click a parameter label, then select **What's This?**

Simulink Examples

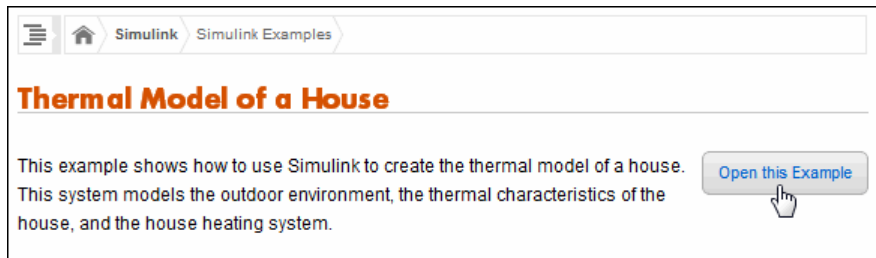
Simulink software provides a variety of example models that illustrate key modeling concepts and Simulink features.

You can access these examples from the **Help** menu or from the documentation.

- From the Simulink Editor, select **Help > Simulink > Examples**.
- From the documentation, click **Examples** at the top of the product landing page.



Featured examples include a button at the top of the Help browser for opening the model in the Simulink Editor.



Web Site Resources

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

To access the Simulink product page, go to

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

Simulink Software Basics

- “Start the Simulink Software” on page 2-2
- “Simulink User Interface” on page 2-6

Start the Simulink Software

In this section...

“Open the Simulink Library Browser” on page 2-2

“Create a New Simulink Model” on page 2-4

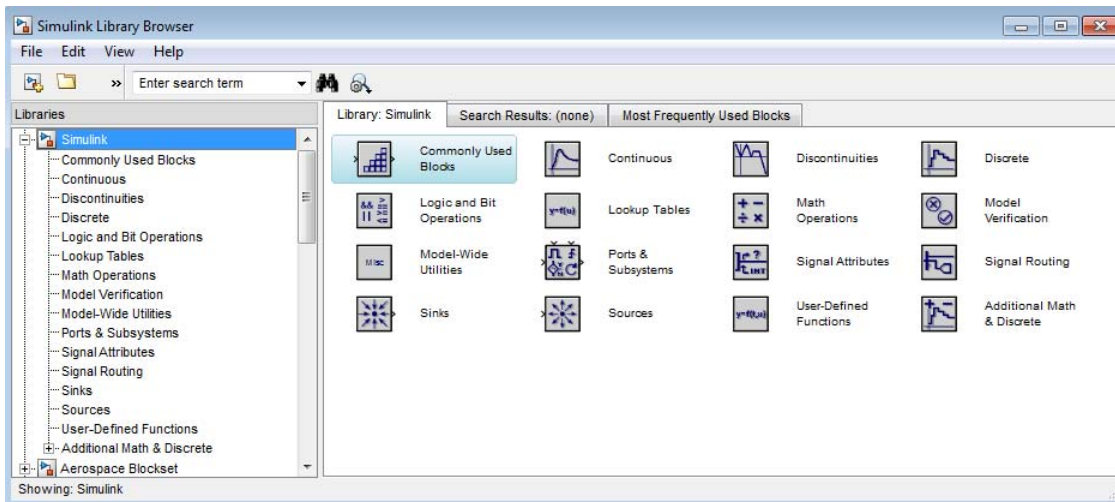
“Open an Existing Model” on page 2-5


Open the Simulink Library Browser

You need MATLAB running before you can open the Simulink Library Browser.

- 1 Start MATLAB, and then in the MATLAB Command Window, enter `simulink`

The Simulink Library Browser opens.



You can also open the Simulink Library Browser from the MATLAB Toolstrip, by clicking the **Simulink Library** button ().

If you have not already loaded Simulink, a short delay occurs while it loads.
The Library Browser opens.

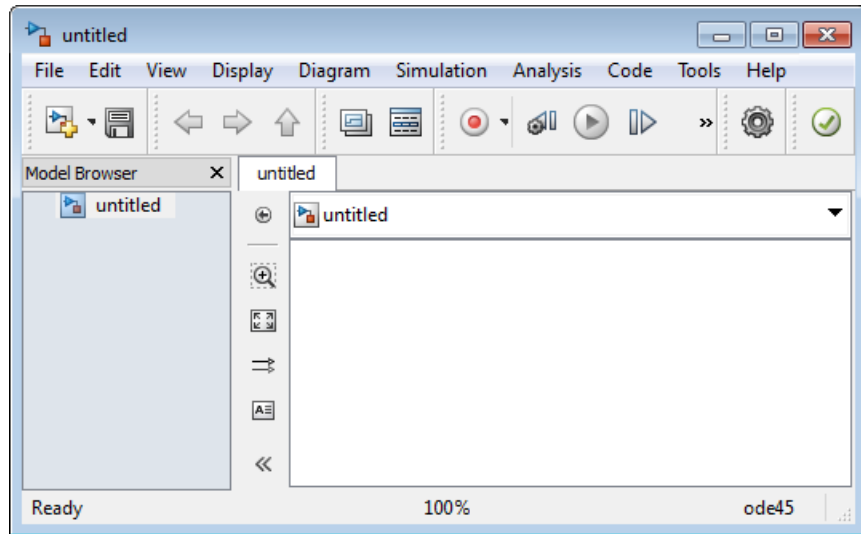
To keep the Library Browser above all other windows on your desktop, in the Library Browser, select **View > Stay on Top**.

Create a New Simulink Model

Create a new Simulink model from the Simulink Library Browser.

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

An empty model opens in the Simulink Editor.



- 2 In the Simulink Editor, select **File > Save**.

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

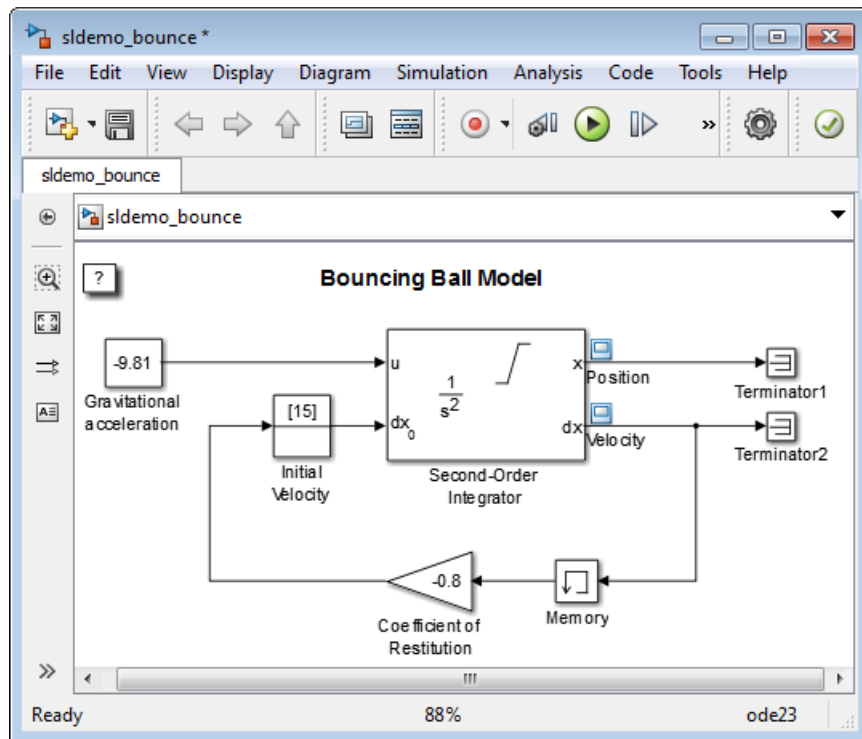
Simulink saves your model.

Open an Existing Model

Open an existing Simulink model from the Simulink Library Browser.

- 1 From the Simulink Library Browser menu, select **File > Open**.
- 2 In the Open dialog box, select the model file that you want to open, and then click **Open**.

The selected model opens in the Simulink Editor. For example:



Alternatively, in MATLAB you can set the current folder to the folder containing your model, and then in the MATLAB Command Window, enter the model name (without the filename extension). For more information about the MATLAB current folder, see “Working with Files and Folders”.

Simulink User Interface

In this section...

“Simulink Library Browser” on page 2-6

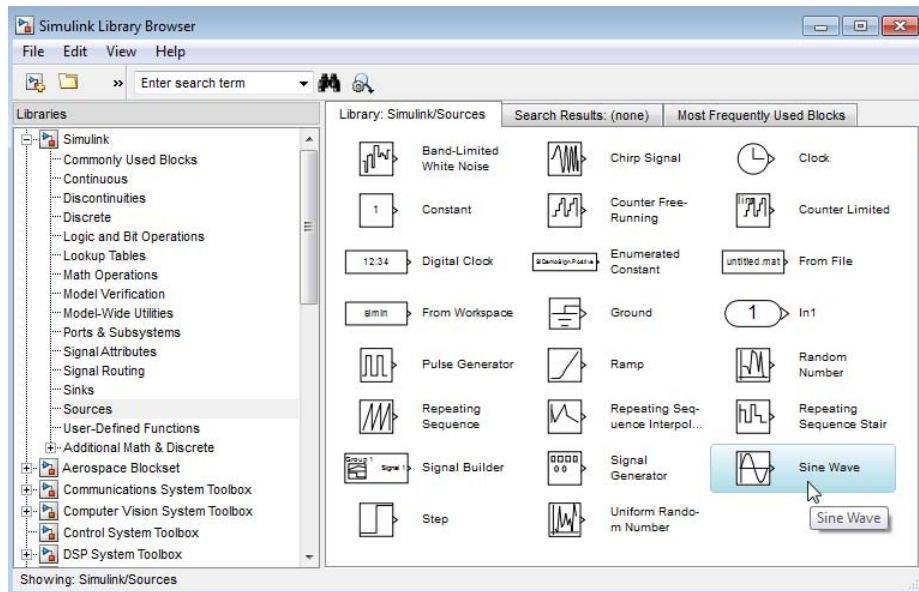
“Simulink Editor” on page 2-9

Simulink Library Browser

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


For example, in the Library Browser below:

- In the **Libraries** pane on the left, the Sources library is the selected library.
- The Sine Wave block is selected.



Tips for Using the Library Browser

You can perform several tasks using the Library Browser, including the following.

Task	Action to Perform Using the Library Browser
View the blocks in a library	In the left pane, select a library name, or in the right pane, double-click a library icon.
Search for a specific block	In the block search field, enter the name of a block, and then click the find block icon  .
Get summary information about a block	Select View > Show Block Descriptions . Then select the block.
Get detailed information about a block	Select the block, and then select Help > Help for the Selected Block . The Help browser opens with the reference page for the block.
View block parameters	Right-click a block, and then select Block parameters .
Copy a block from the Library Browser to a model	Drag a block from the Library Browser to a Simulink Editor model window.

Standard Block Libraries

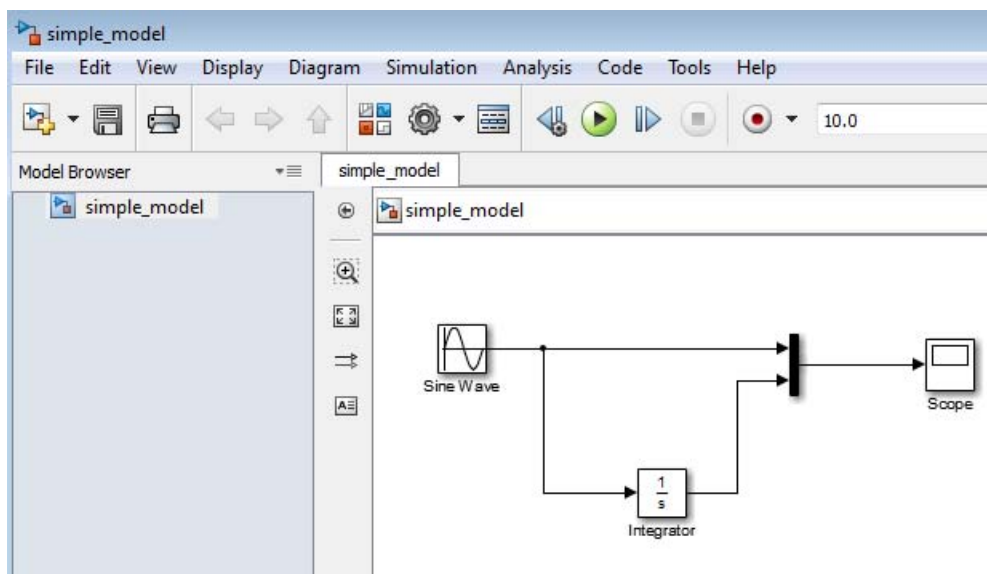
Simulink software includes the following standard block libraries.

Block Library	Description
Commonly Used Blocks	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	Model linear functions using blocks such as the Derivative and Integrator blocks.

Block Library	Description
Discontinuities	Create outputs that are discontinuous functions of their inputs using blocks such as the Saturation block.
Discrete	Represent discrete time functions using blocks such as the Unit Delay block.
Logic and Bit Operations	Perform logic or bit operations using blocks such as the Logical Operator and Relational Operator blocks.
Lookup Tables	Use lookup tables to determine their outputs from their inputs using blocks such as the Cosine and Sine blocks.
Math Operations	Perform mathematical and logical functions using blocks such as the Gain , Product , and Sum blocks.
Model Verification	Create self-validating models using blocks such as the Check Input Resolution block.
Model-Wide Utilities	Provide information about the model using blocks such as the Model Info block.
Ports & Subsystems	Create subsystems using blocks such as the In1 , Out1 , and Subsystem blocks.
Signal Attributes	Modify the attributes of signals using blocks such as the Data Type Conversion block.
Signal Routing	Route signals from one point in a block diagram to another using blocks such as the Mux and Switch blocks.
Sinks	Display or export output using blocks such as the Out1 and Scope blocks.
Sources	Generate or import system inputs using blocks such as the Constant , In1 , and Sine Wave blocks.
User-Defined Functions	Define custom functions using blocks such as the MATLAB Function block.
Additional Math & Discrete	Additional libraries for mathematical and discrete function blocks.

Simulink Editor

The Simulink Editor contains a block diagram of your model. You build models by dragging blocks from the Simulink Library Browser window to the Simulink Editor model window. In the model window, you build a block diagram by arranging the blocks logically, connecting the blocks with signal lines, and setting the parameters for each block.



You also use the Simulink Editor 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.

Create a Simulink Model

Create a Simple Model

In this section...
“Overview of the Simple Model” on page 3-2
“Create the Simple Model” on page 3-3
“Connect Blocks in the Simple Model” on page 3-8
“Simulate the Simple Model” on page 3-13

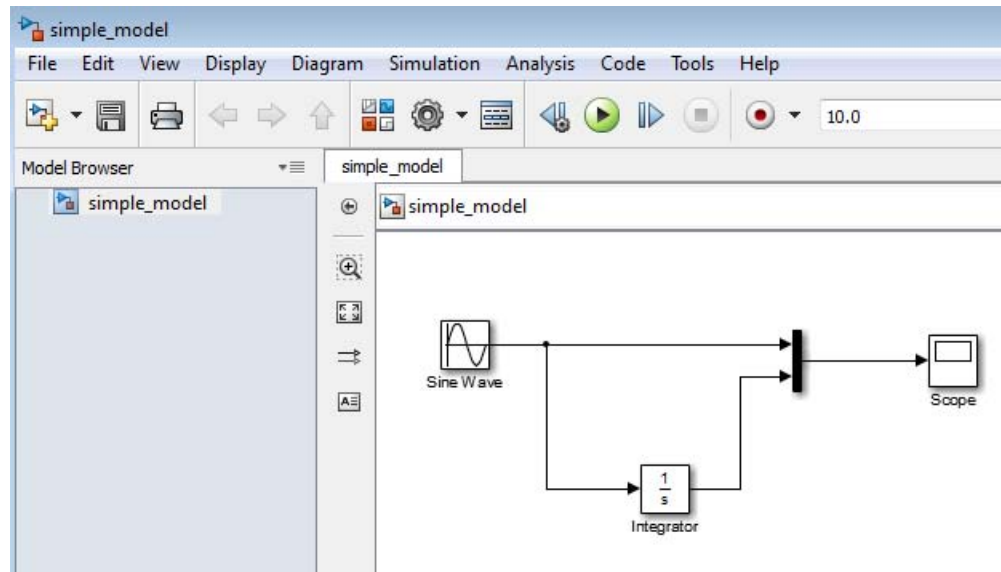
Overview of the Simple Model

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

To create this simple model, you need four blocks:

- **Sine Wave** — Generates an input signal for the model.
- **Integrator** — Processes the input signal.
- **Mux** — Combines the input signal and processed signal into one signal.
- **Scope** — Visualizes the signals.

After you connect the blocks, simulating the model integrates a sine wave signal and displays the result, along with the original signal.



You can build this simple model yourself, starting with “Create a New Model” on page 3-3.

Create the Simple Model

- “Create a New Model” on page 3-3
- “Add Blocks to a Model” on page 3-4
- “Move Blocks in the Model” on page 3-7

Create a New Model

Before creating a model, you need to start MATLAB and then start Simulink.

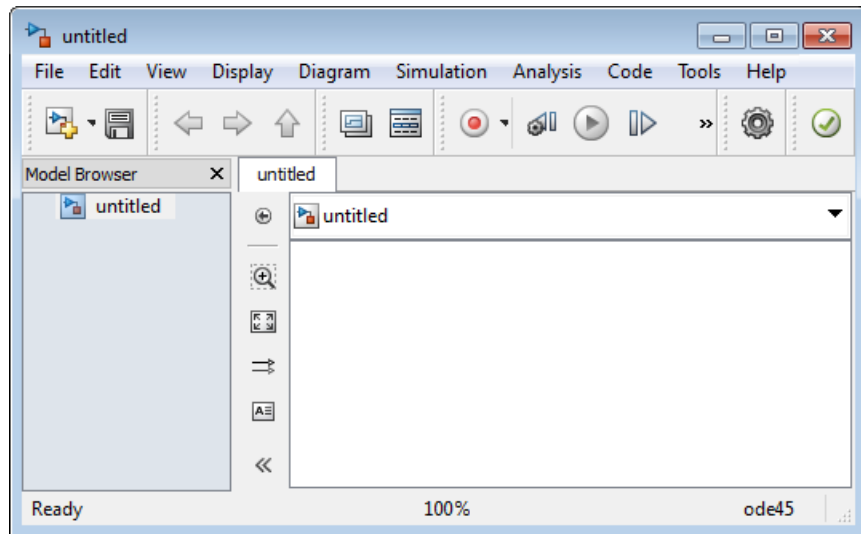
- 1 Start the Simulink software. In the MATLAB Command Window, enter:

```
simulink
```

The Simulink Library Browser opens.

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

A Simulink Editor window opens with an empty canvas in the right-hand pane (the model window).



3 Select **File > Save as**. The Save As dialog box opens.

4 In the **File name** box, enter a name for your model. For example, enter `simple_model`. Then click **Save**.

Your model is saved with the file name `simple_model`.

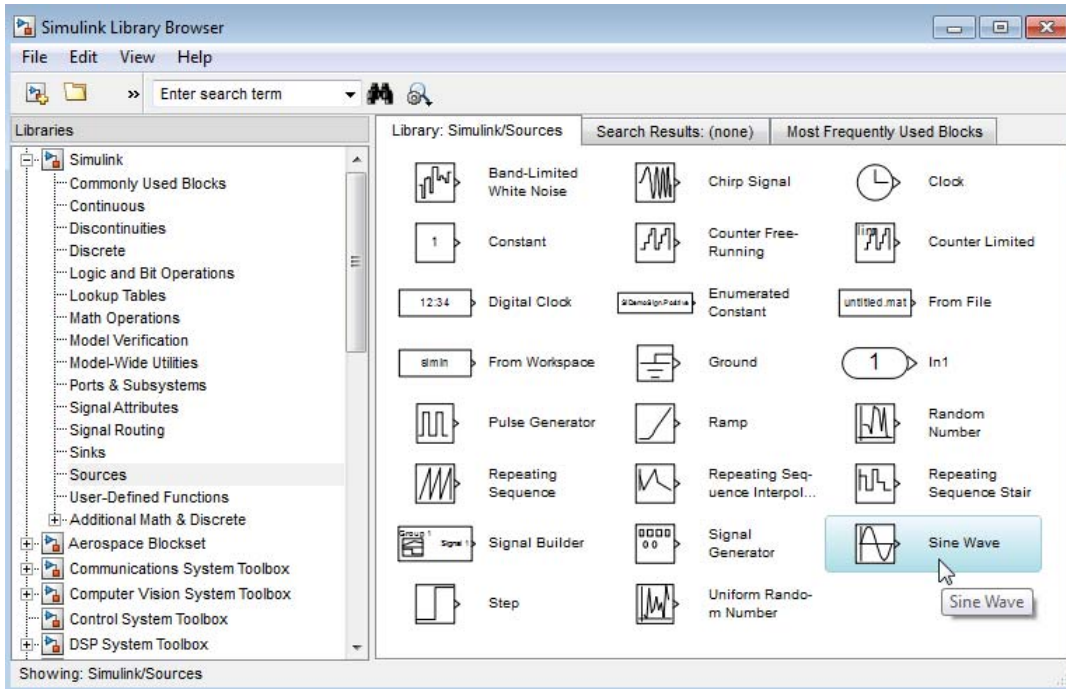
Add Blocks to a Model

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

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

The Simulink Library Browser displays blocks from the Sources library in the right-hand pane.

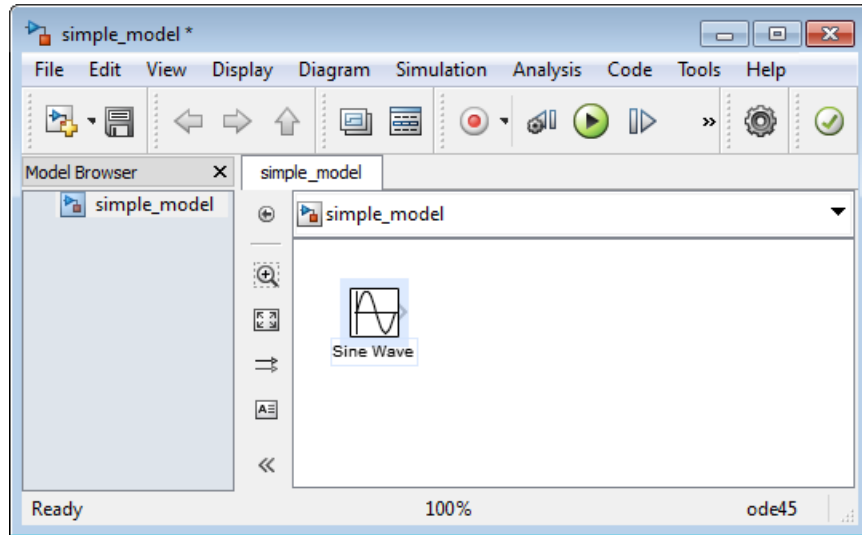
2 Select the Sine Wave block.



Drag the Sine Wave block to the model window.

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

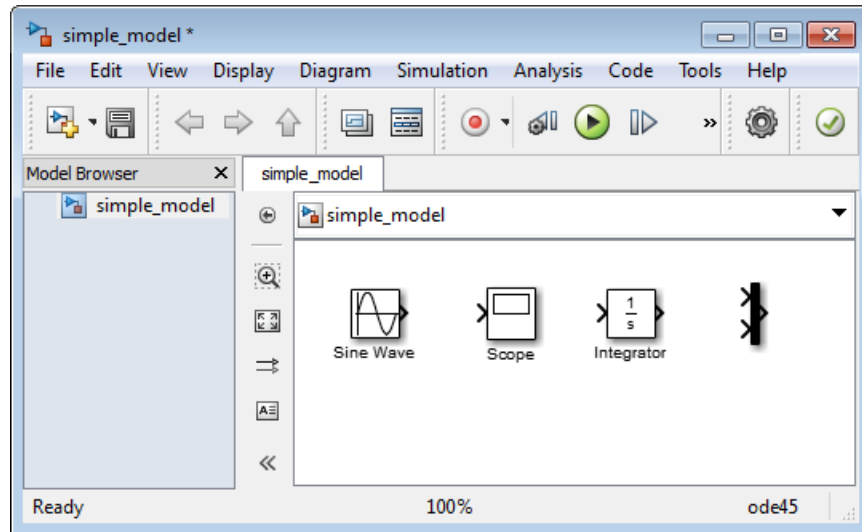
3 Create a Simulink® Model



- 3 Add the following blocks to your model, using the same approach that you used to add the **Sine Wave** block.

Library	Block
Sinks	Scope
Continuous	Integrator
Signal Routing	Mux

Your model now has the blocks you need for the simple model.

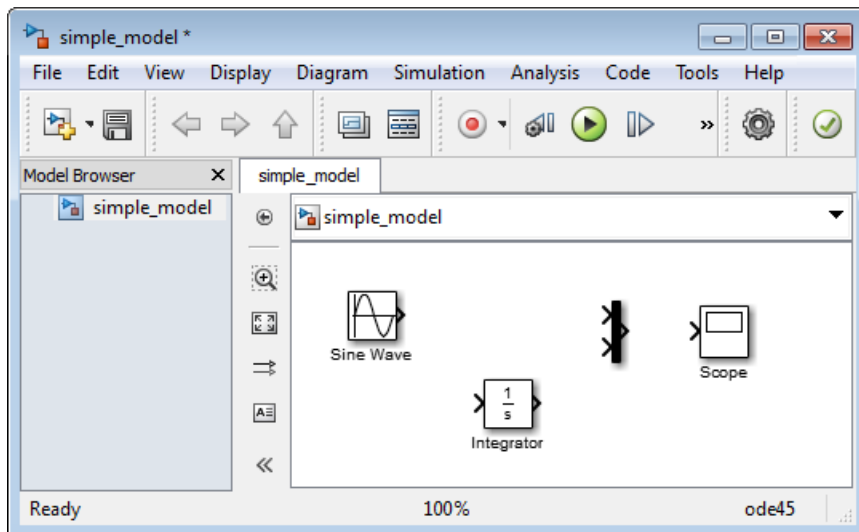


Move Blocks in the Model

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

- 1 Move the Scope block after the Mux block output. To move a block in a model, use one of these approaches:
 - Click and drag the block.
 - Select the block, and then press the arrow keys on the keyboard.
- 2 Move the Sine Wave and Integrator blocks before the Mux block.

Your model should look similar to the following figure.



Your next task is to connect the blocks with signal lines.

Connect Blocks in the Simple Model

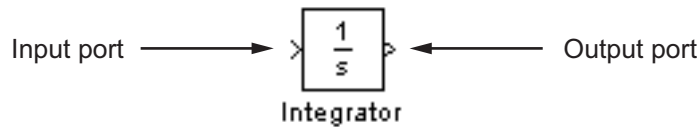
- “Block Connections in a Model” on page 3-8
- “Draw Lines Between Blocks” on page 3-9
- “Draw a Branch Line” on page 3-11

Block Connections in a Model

After you add blocks to your model, you need to connect them. The connecting lines represent the signals within a 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*.



Draw Lines Between Blocks

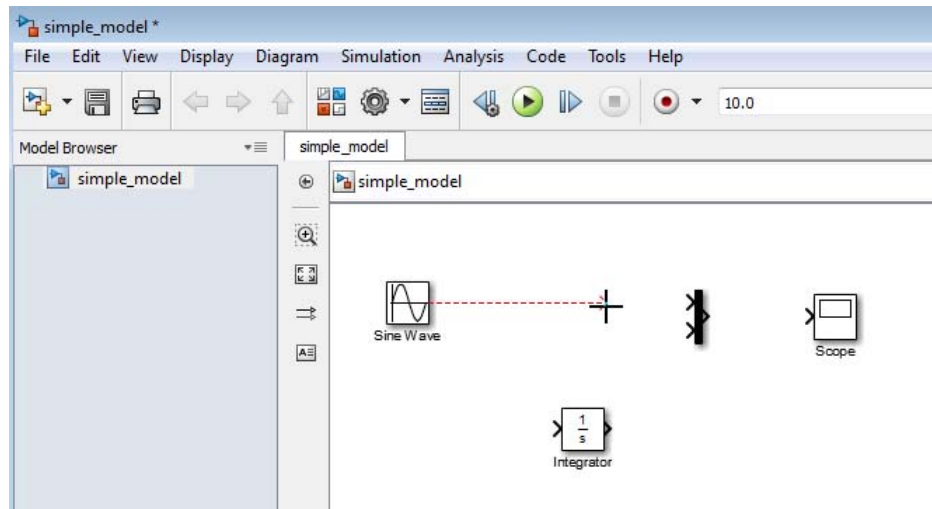
Connect the blocks by drawing lines between output ports and input ports. For information about how to add blocks to the model in this example, see “Add Blocks to a Model” on page 3-4.

- 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 Click and drag a line from the output port to the top input port of the Mux block.

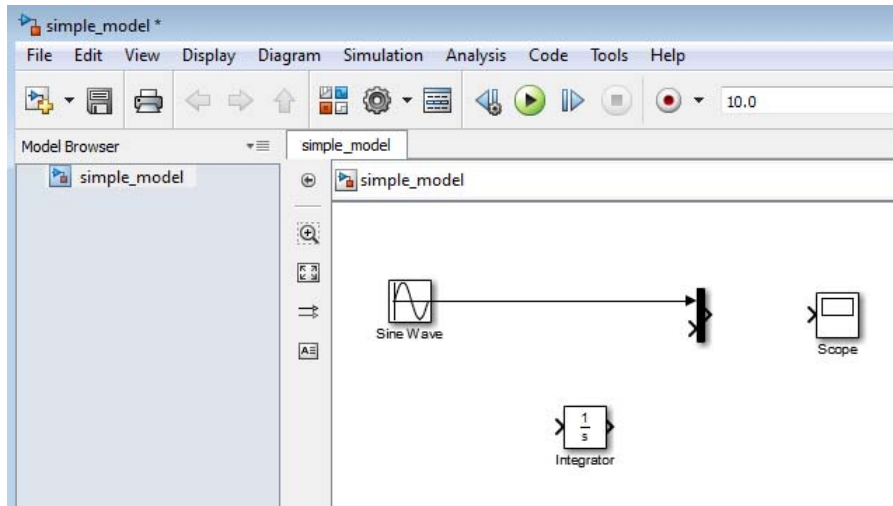
While you hold the mouse button down, the connecting line appears as a red dotted arrow.



- 3 Release the mouse button over the output port.

3 Create a Simulink® Model

Simulink connects the blocks with an arrow indicating the direction of signal flow.

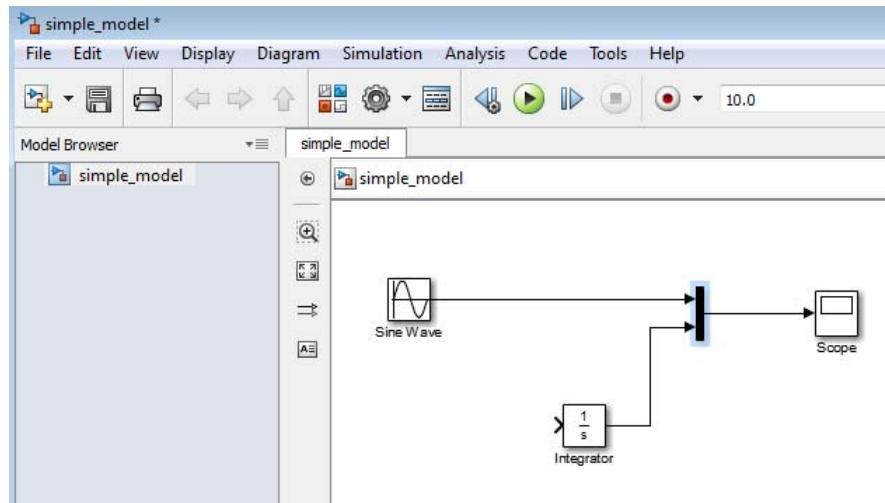


- 4 Connect the output port of the Integrator block to the bottom input port on the Mux block:
 - a Select the Integrator block.
 - b Press and hold the **Ctrl** key.
 - c Select the Mux block.

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

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

- 5 Connect the Mux block output port to the Scope block.



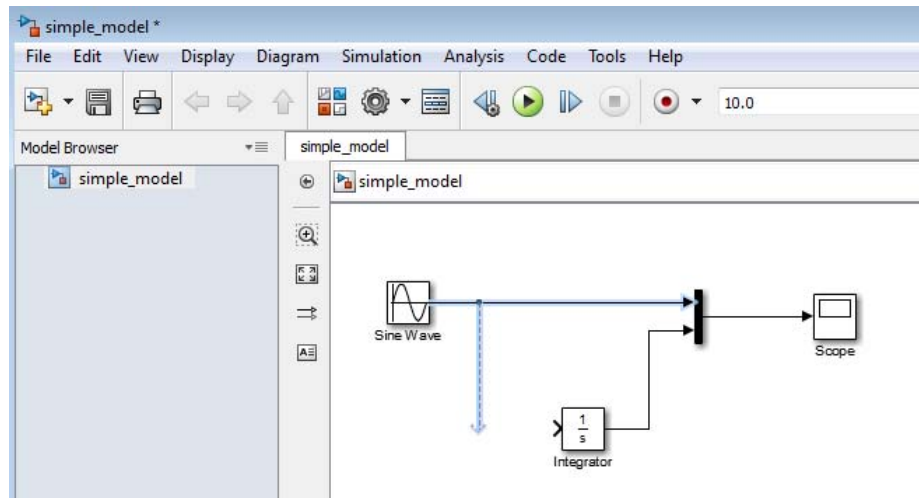
Draw a Branch Line

The simple model is almost complete, except for one connection. To finish the model, you need to connect the Sine Wave block to the Integrator block.

This final connection is different from the other three connections, which all connect output ports to input ports. Because the output port of the Sine Wave block is already connected, 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. Click the left mouse button.
- 2 Hold down the **Ctrl** key and drag the cursor to form a vertical line segment that aligns with input port of the Integrator block.

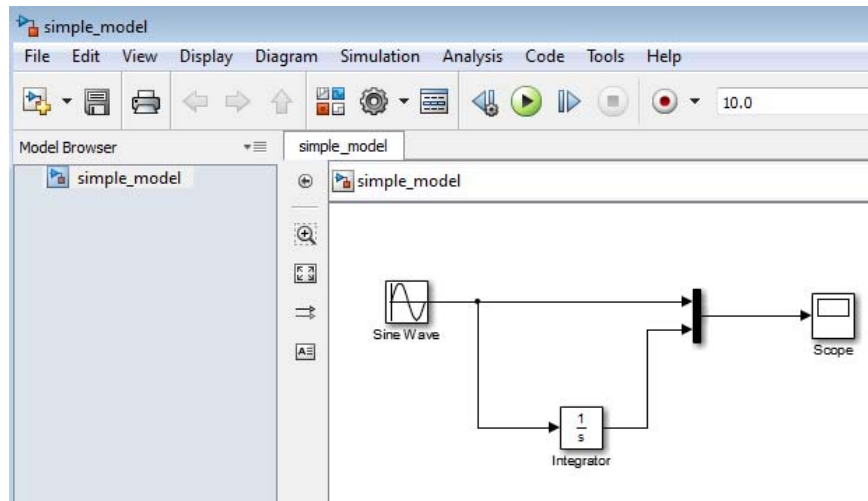
3 Create a Simulink® Model



- 3 With the cursor at the end of the dotted red line, move the cursor so that it lines up with the light blue guide arrow that points in the direction of the Integrator input port. The cursor turns into a hollow arrow.

Drag the cursor to the Integrator input port and release the mouse button.

- 4 If necessary, smooth out the line segment that connects to the Integrator block. Select the unaligned line segment and drag it until the line is straight.
- 5 Your model diagram is now complete. Select **File > Save**.



After you finish building a model, you can simulate the dynamic behavior of the model. See “Simulate the Simple Model” on page 3-13.

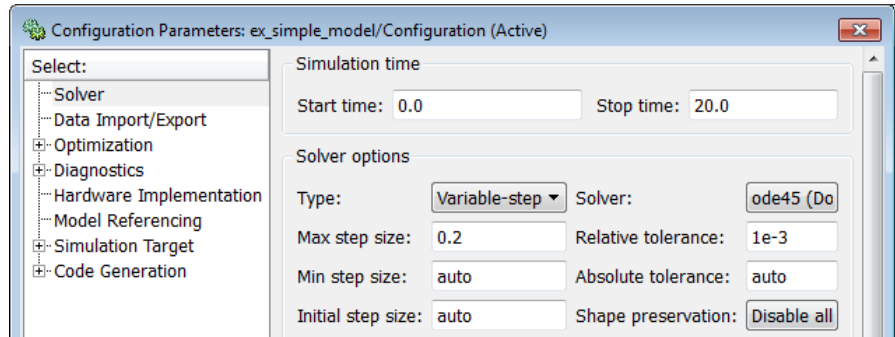
Simulate the Simple Model

- “Set Simulation Options” on page 3-13
- “Run a Simulation and Observe Results” on page 3-14

Set Simulation Options

Before you simulate a model, you have to set simulation options. Specify options, such as the stop time and solver, using the Model Configuration Parameters dialog box. For how to build the model in this example, see “Create the Simple Model” on page 3-3.

- 1** In the Simulink Editor, select **Simulation > Model Configuration Parameters**. The Configuration Parameters dialog box opens to the **Solver** pane.
- 2** In the **Stop time** field, enter 20. In the **Max step size** field, enter 0.2.



3 Click **OK**.

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



Run a Simulation and Observe Results

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

1 In the Simulink Editor, select **Simulation > Run**.

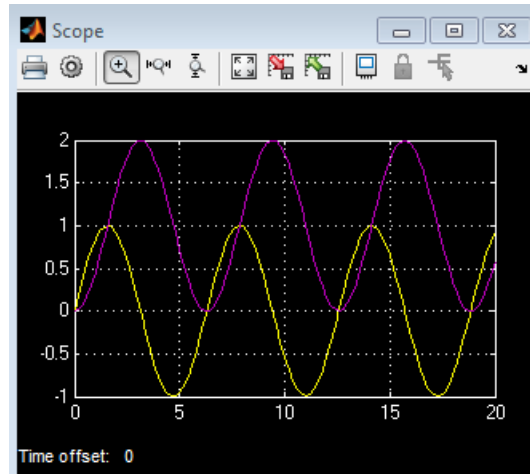
The simulation runs. The simulation stops when it reaches the stop time specified in the Configuration Parameters dialog box.


Tip Alternatively, you can control a simulation by clicking the **Run**

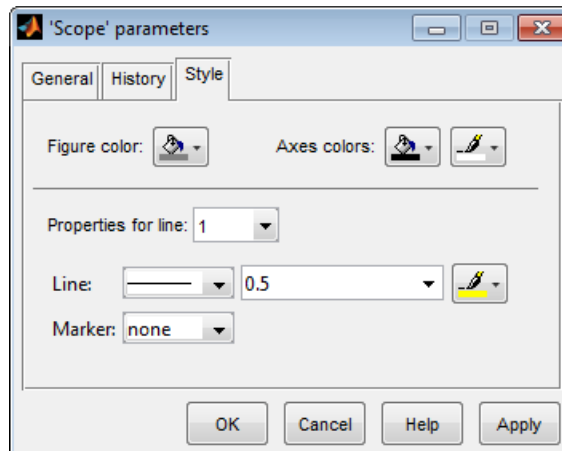
simulation button  and **Pause** simulation button  on the Simulink Editor toolbar.

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.

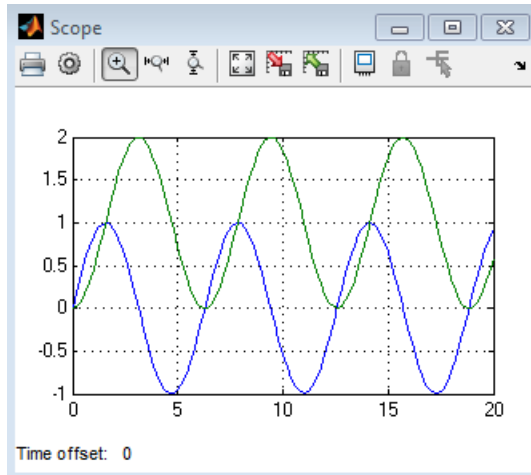


- 3 From the Scope block toolbar, click the **Parameters** button . Select the **Style** tab. The Scope Parameters dialog box displays figure editing options.



- 4 Change the appearance of the figure. For example, select white for the Figure color and Axes background color (icon with a pitcher) and black for the Ticks, labels and grid colors (icon with a paintbrush). Change the signal line colors for signal 1 to blue and for signal 2 to green. To see your changes, click **OK**.

3 Create a Simulink® Model



5 Select **File > Close > Close Model**.

The Simulink Editor closes with changes to your model and model configuration parameters.

Modeling a Dynamic Control System

- “Dynamic Control System Model” on page 4-2
- “Anatomy of the Control System Model” on page 4-3
- “Simulate the Control System Model” on page 4-14
- “Move Data Between MATLAB and Simulink Model” on page 4-21

Dynamic Control System Model

This tutorial uses the `sldemo_househeat` model to illustrate how you can use Simulink software to model a dynamic control system. The model defines a heating system and the thermodynamics of a house. It includes the outdoor environment, the thermal characteristics of a house, and the house heating system.

Use this tutorial to explore common Simulink modeling tasks, such as:

- Simplify a block diagram by grouping multiple blocks into a single *subsystem* block.
See “Anatomy of the Control System Model” on page 4-3.
- Simulate a model, observe results, and change input parameters to investigate how the system responds.
See “Simulate the Control System Model” on page 4-14.
- Import data from the MATLAB workspace to the model, and export runtime data back to MATLAB.
See “Move Data Between MATLAB and Simulink Model” on page 4-21.

Anatomy of the Control System Model

In this section...

“Open the Example Model” on page 4-3

“Overview of the Example Model” on page 4-4

“Subsystems in the Example Model” on page 4-6

“Subsystems and Masks” on page 4-10

“Creating a Subsystem” on page 4-11

“Create a Subsystem Mask” on page 4-12

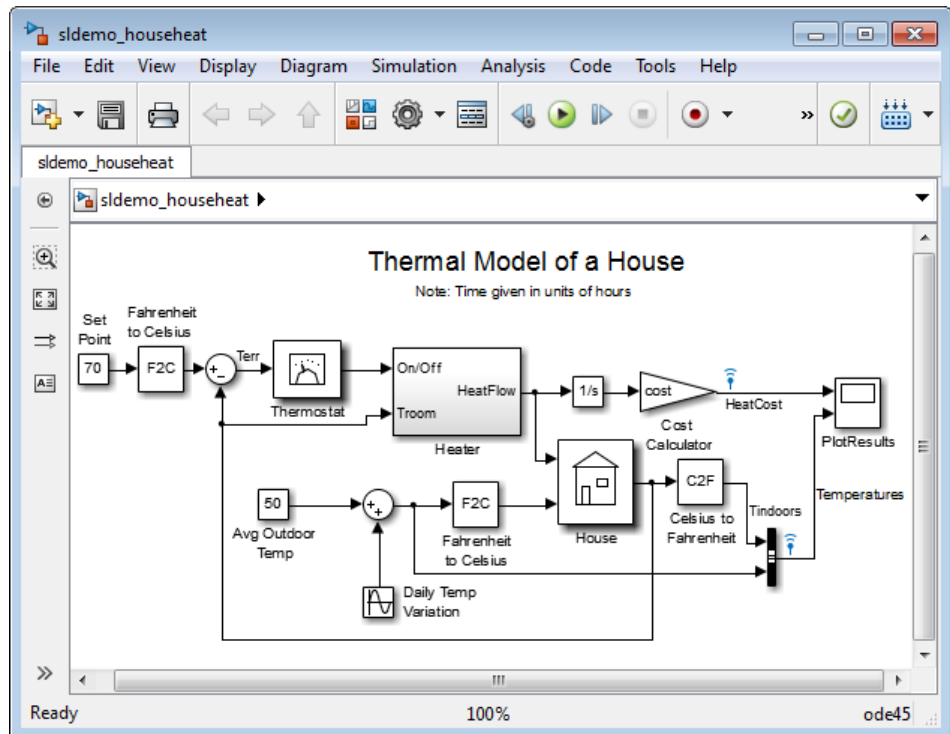
Open the Example Model

The model for this example is called `sldemo_househeat`. It models the heating system and thermodynamics of a house.

1 Start MATLAB, and then in the MATLAB Command Window, enter

```
sldemo_househeat
```

The Simulink Editor opens with the example model.



Overview of the Example Model

The `sldemo_househeat` 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 example model includes many of the same blocks you used to create the simple model in “Create a Simple Model” on page 3-2. These blocks 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 model, the thermostat is set to 70 degrees Fahrenheit. Fluctuations in outdoor temperature are modeled 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 Example Model

A *subsystem* is a hierarchical grouping of blocks encapsulated by a single Subsystem block. The `sldemo_househeat` model uses subsystems to:

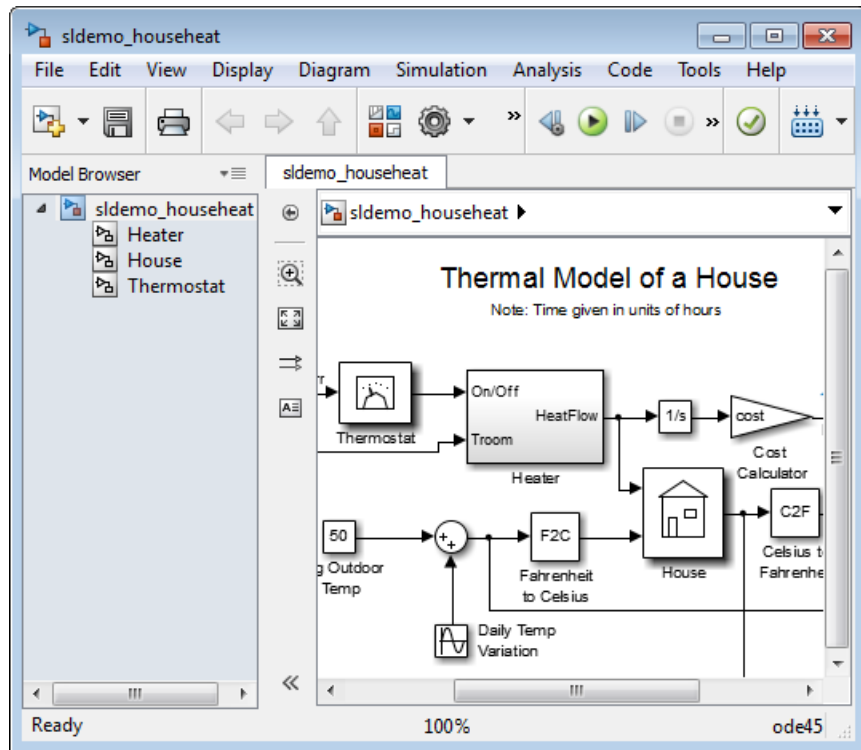
- Simplify the appearance of the block diagram
- Create reusable components
- Customize the appearance of blocks

The example model uses the following subsystems: Thermostat, Heater, House, Fahrenheit to Celsius, and Celsius to Fahrenheit.

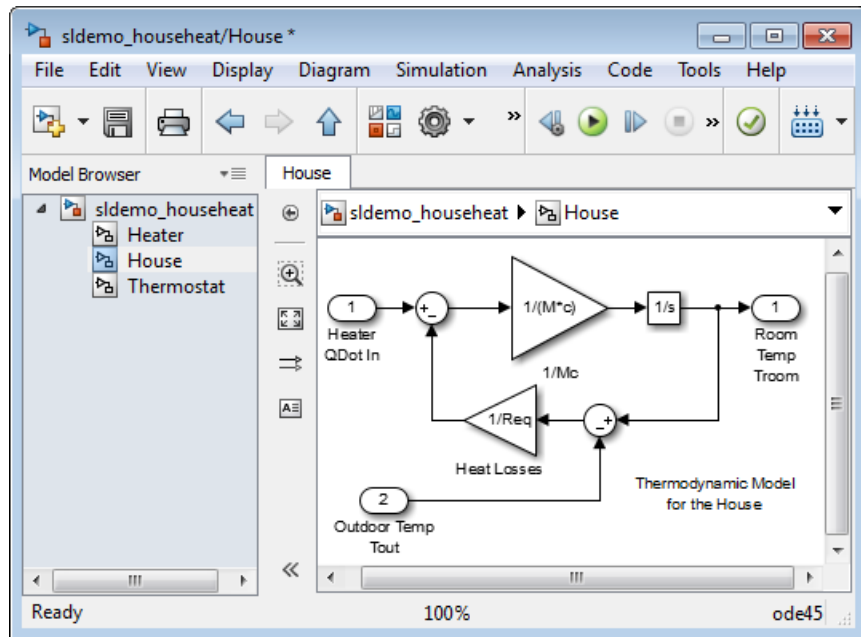
1 In the MATLAB Command Window, enter

```
sldemo_househeat
```

The example model opens in the Simulink Editor model window.

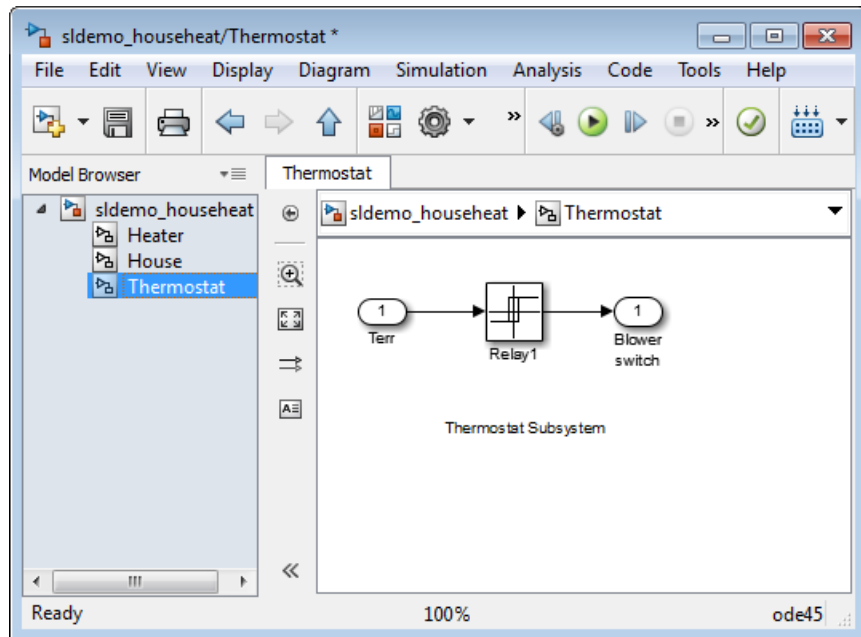


- Subsystems can be complex and contain many blocks that might otherwise clutter a diagram. For example, to open the House subsystem, double-click the House Subsystem block.



The House 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 at the top level of the main `sldemo_househeat` model, but combining them as a subsystem helps to simplify the block diagram.

- 3** A subsystem can also be simple and contain only a few blocks. For example, to open the Thermostat subsystem:
 - a** Display the `sldemo_househeat` top model. To do so, in the breadcrumb at the top of the model window, (`sldemo_househeat` ▶ `House`), click `sldemo_househeat`. The `sldemo_househeat` model appears in the model window.
 - b** Double-click the Thermostat Subsystem block.

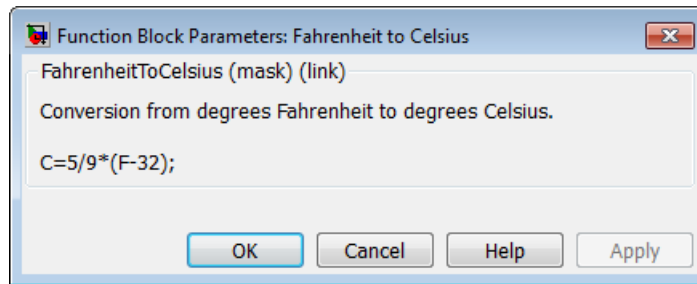


This subsystem models the operation of a thermostat, determining when the heating system is on or off. It contains only one Relay block, but logically represents the thermostat in the block diagram.

Subsystems and Masks

The `sldemo_househeat` 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.

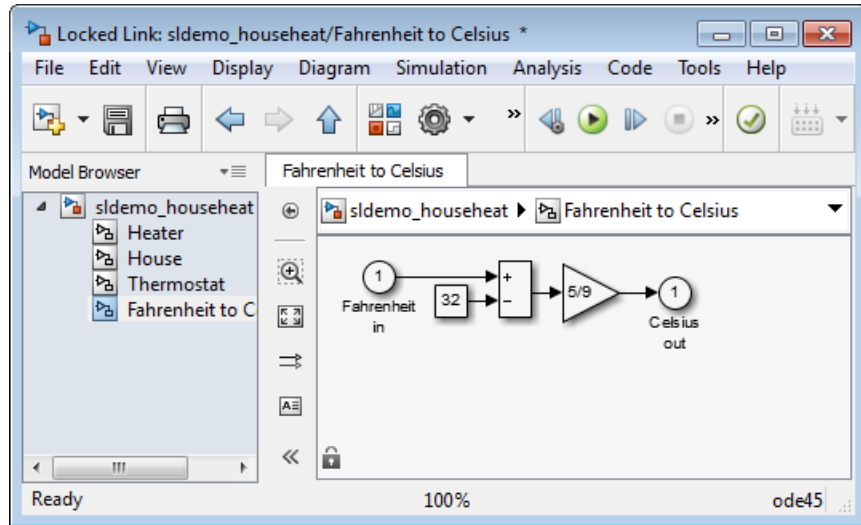
- 1 Display the `sldemo_househeat` model, by selecting `sldemo_househeat` in the breadcrumb at the top of the Simulink Editor model window.
- 2 Double-click the `Fahrenheit to Celsius` masked Subsystem block. The custom dialog box for the F2C block opens.



Click **OK**.

- 3 To view the underlying blocks in the `Fahrenheit to Celsius` conversion subsystem, select the `Fahrenheit to Celsius` Subsystem block and then select **Diagram > Mask > Look Under Mask**.

The Simulink Editor displays the blocks behind the mask.

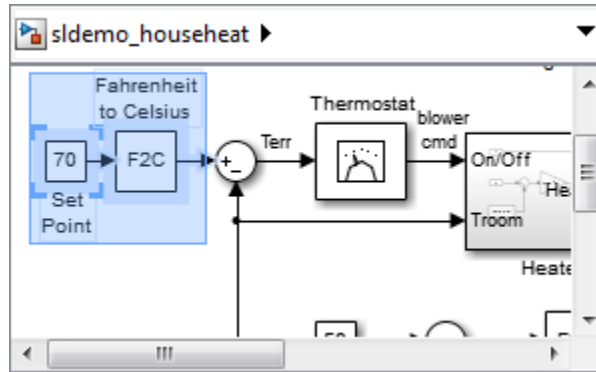


Creating a Subsystem

Subsystems allow you to group related blocks into one block. They are also reusable, enabling you to implement an algorithm once and use it multiple times.

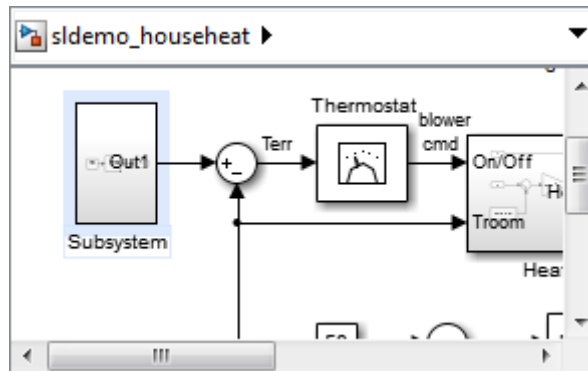
To create a subsystem:

- 1 Display the `sldemo_househeat` model, by selecting `sldemo_househeat` in the breadcrumb at the top of the Simulink Editor model window.
- 2 In the model window, click and drag a bounding box around the Set Point and Fahrenheit to Celsius blocks.



- 3 From the Simulink Editor menu, select **Diagram > Subsystem & Modeling Reference > Create Subsystem from Selection**.

The blocks are combined into one subsystem block.



For more information about working with subsystems, see “Create a Subsystem”.

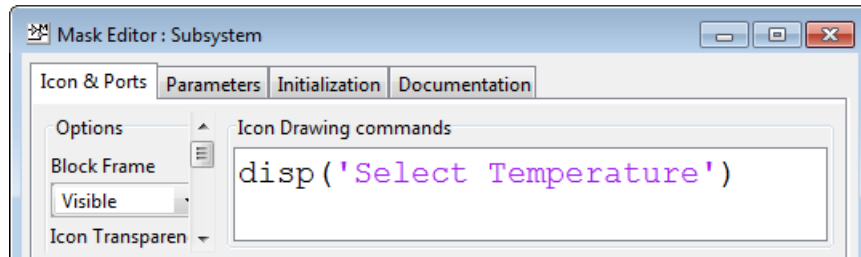
Create a Subsystem Mask

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 the blocks.

To mask a subsystem:

- 1** In the model window, right-click the new subsystem block (as described in “Creating a Subsystem” on page 4-11). In the context menu, select **Mask > Create Mask**. The Mask Editor dialog box opens.
- 2** At the bottom of the Mask Editor dialog box, expand the **Command** list. Select `disp (show text in center of block)`. The dialog box displays the syntax for this command below the list.
- 3** In the **Icon Drawing commands** field, enter `disp('Select Temperature')`.



- 4** Click **OK**. The subsystem block displays the text that you entered.

To see the full text in the block icon, select a corner of the block and drag to expand the size of the block.

For more information about masking subsystems, see “Masking”.

Simulate the Control System Model

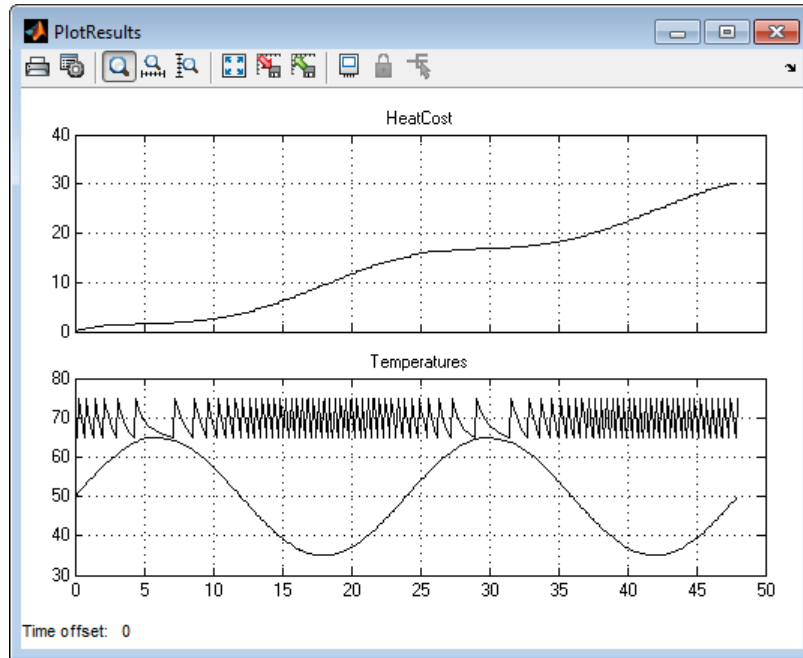
In this section...
“Run the Simulation” on page 4-14
“Change the Thermostat Setting” on page 4-15
“Change the Average Outdoor Temperature” on page 4-17
“Change the Daily Temperature Variation” on page 4-19

Run the Simulation

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

- 1 In the `sldemo_househeat` model window, double-click the Scope block named `PlotResults`. A Scope window opens with two axes labeled `HeatCost` and `Temperatures`.
- 2 In the Simulink Editor, select **Simulation > Run**.

The software simulates the model dynamics, and as the simulation runs, it displays the results in a Scope window. The cumulative heating cost is drawn on the `HeatCost` graph. The indoor and outdoor temperatures appear on the `Temperatures` graph.

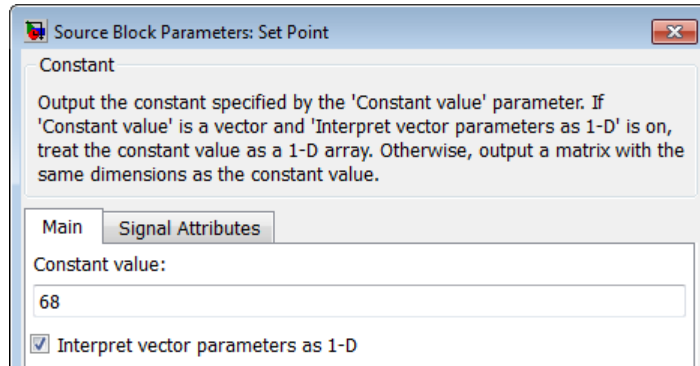



Change the Thermostat Setting

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.

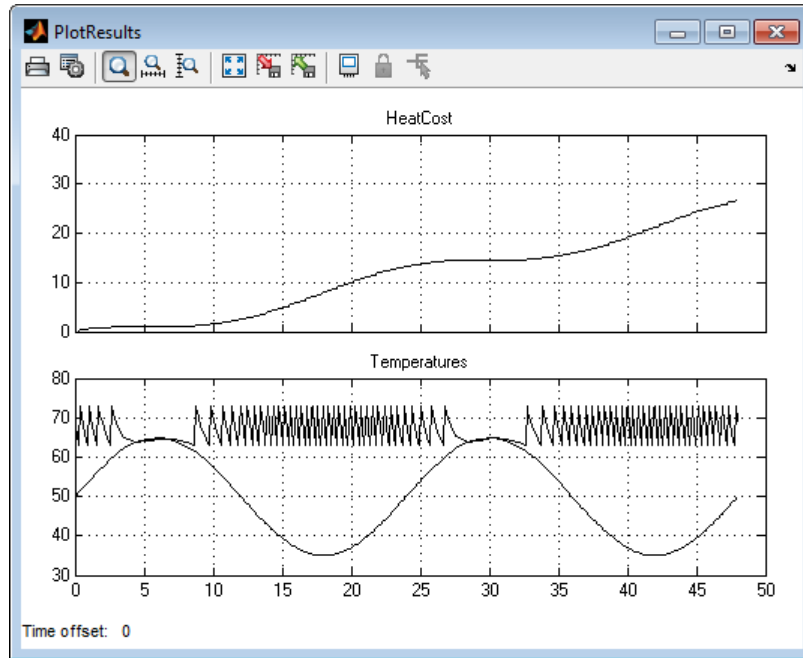
Change the thermostat setting to 68 degrees Fahrenheit and observe how the model responds.

- 1** In the `sldemo_househeat` model window, open the masked Subsystem block (the block icon that displays the text `Select Temperature`).
- 2** In the Subsystem model window, double-click the `Set Point` block. The `Source Block Parameters` dialog box opens.
- 3** In the **Constant value** field, enter `68`.



- 4 Click **OK**. The software applies your changes and closes the dialog box.
- 5 To rerun the simulation, you can either select **Simulation > Run**, or from the toolbar, you can use the **Run** button ().

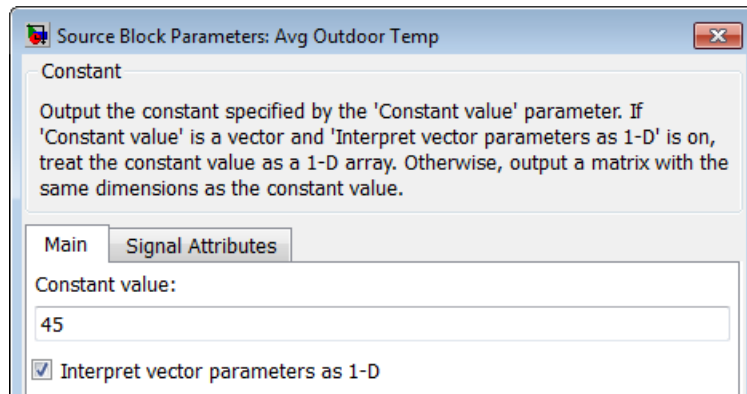
The software simulates the model dynamics. Notice that a lower thermostat setting reduces the cumulative heating cost.



Change the Average Outdoor Temperature

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

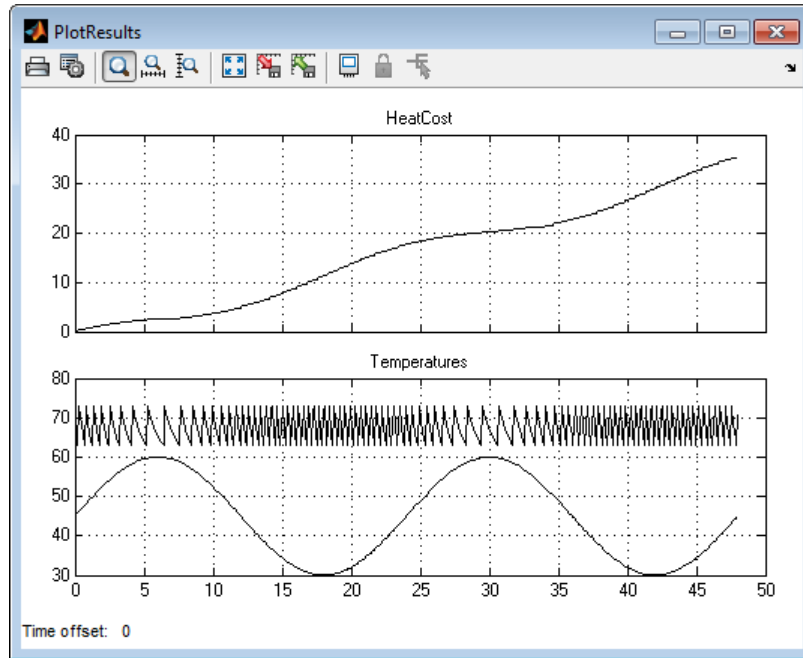
- 1 In the `sldemo_househeat` model window, double-click the Avg Outdoor Temp block. The Source Block Parameters dialog box opens.
- 2 In the **Constant value** field, enter 45.



3 Click **OK**. The software applies your changes.

4 Rerun the simulation.

The software simulates the model dynamics. Notice that a colder outdoor temperature increases the cumulative heating cost.



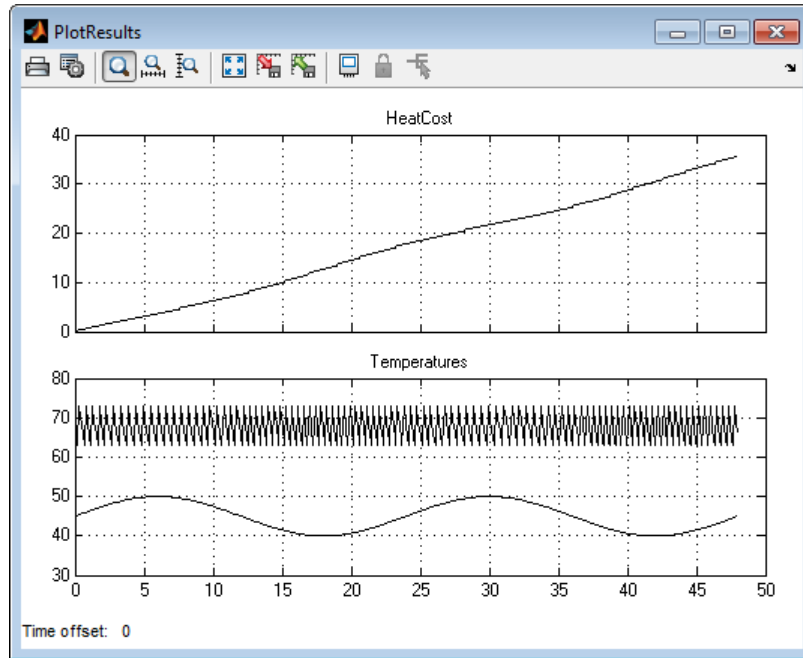
Change the Daily Temperature Variation

Decrease the temperature variation to see how the model responds.

- 1** In the `sldemo_househeat` model window, double-click the Daily Temp Variation block. The Source Block Parameters dialog box opens.
- 2** In the **Amplitude** field, enter 5.
- 3** Click **OK**. The software applies your changes.
- 4** Rerun the simulation.

The software simulates the model dynamics. Notice that a more stable outdoor temperature alters the frequency with which the heater operates.

4 Modeling a Dynamic Control System



Move Data Between MATLAB and Simulink Model

In this section...

“Import Data from the MATLAB Workspace” on page 4-21

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

Import Data from the MATLAB Workspace

You can import data from the MATLAB workspace to the Simulink model input ports. This allows you to import actual physical data into your model. For information about other data import capabilities, see “Import Data”.

In this example, you 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:

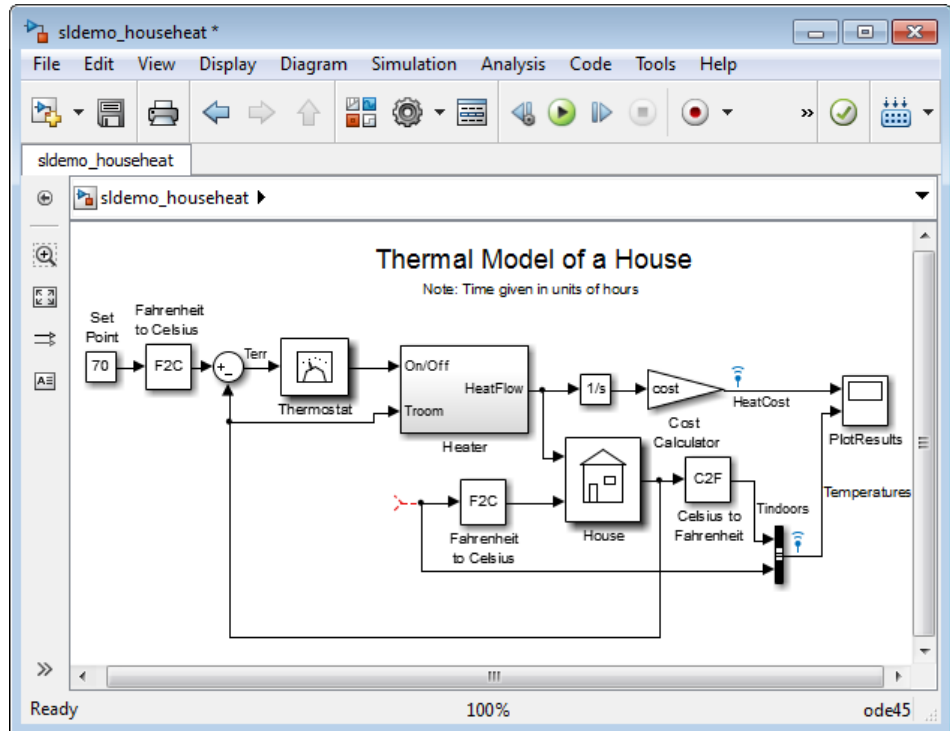
- 1 In the MATLAB Command Window, create time and temperature data by entering the following commands:

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

y is a vector of outdoor temperatures that varies from 27 to 37 degrees Fahrenheit. z is a time vector from 0 to 48. Both vectors contain 1257 values.

- 2 In the Simulink Editor model window, select the Avg Outdoor Temp block, and then press the **Delete** key to remove it from the model.
- 3 Delete the following items from the model in the same way:
 - Daily Temp Variation block
 - Two input signal lines to the Sum1 block
 - Sum1 block

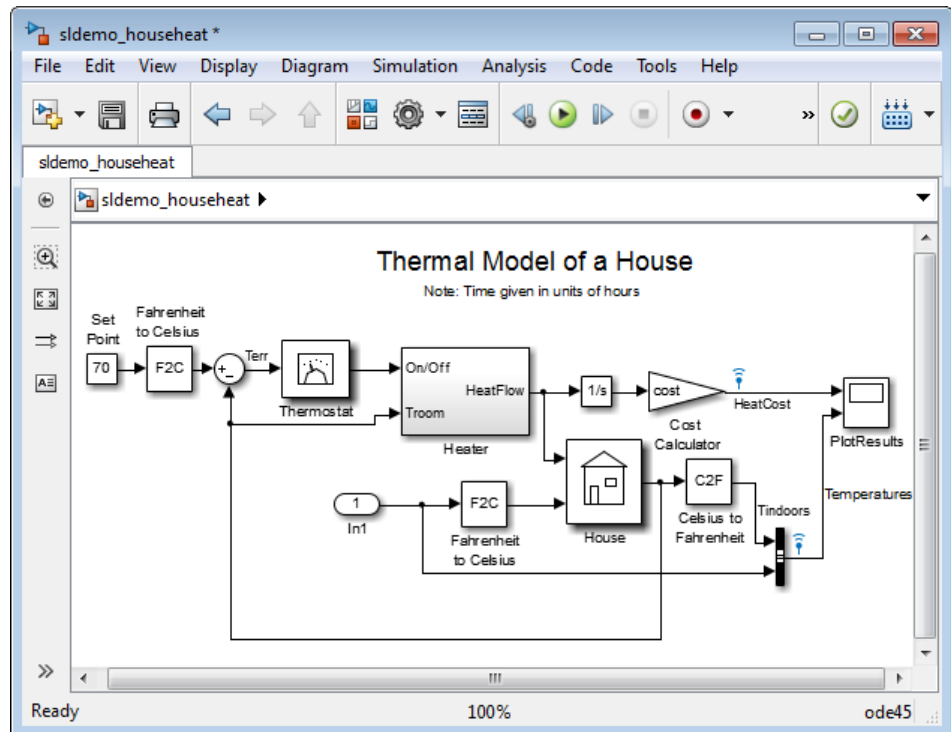
The model should now look similar to the following figure. Notice that the output signal from the Sum1 block changes to a red, dotted line, indicating that it is not connected to a block.



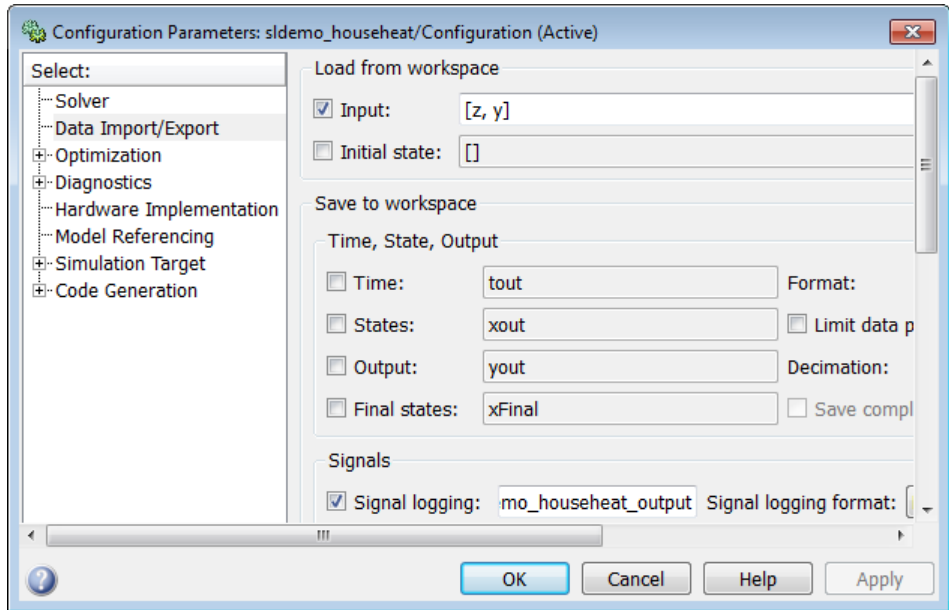
- 4 In the model window, select **View > Library Browser**. The Simulink Library Browser window opens.
- 5 In the Library Browser, in the **Libraries** pane, select the Sources library.
- 6 From the Sources library in the right pane, select the In1 block, and then drag it to the Simulink Editor model window.

An instance of the In1 block appears in the model window.

- 7 Connect the dotted line (originally connected to the Sum1 block) to the In1 block.

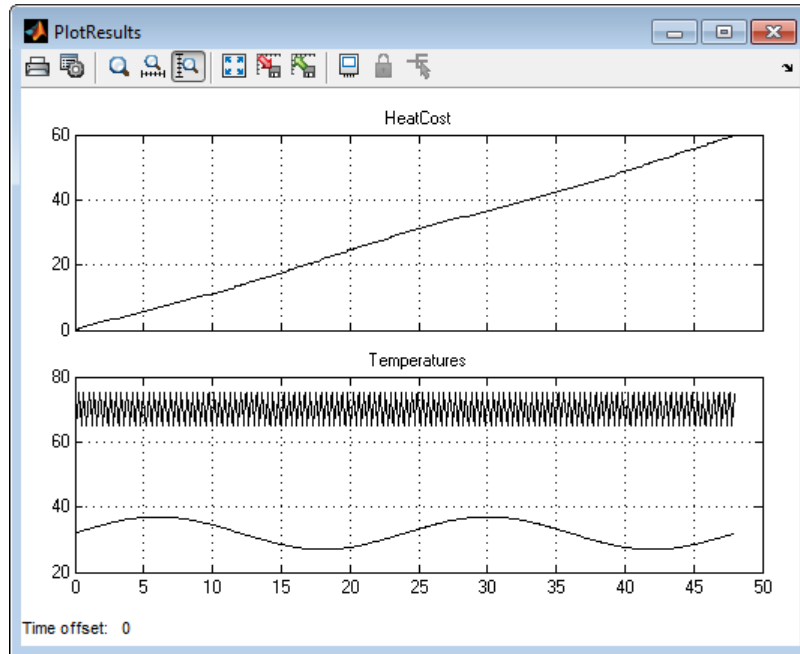


- 8** In the Simulink Editor, select **Simulation > Model Configuration Parameters**.
- 9** From the menu on the left side of the Configuration Parameters dialog box, select **Data Import/Export**.
- 10** On the **Data Import/Export** pane, select the **Input** check box. In the **Input** field, enter $[z, y]$.



- 11** Click **OK**. The software applies your changes and closes the dialog box.
- 12** Rerun the simulation.

The software simulates the model. Notice that the model ran using the imported data, showing colder outdoor temperatures and higher heat use.



Export Simulation Data to the MATLAB Workspace

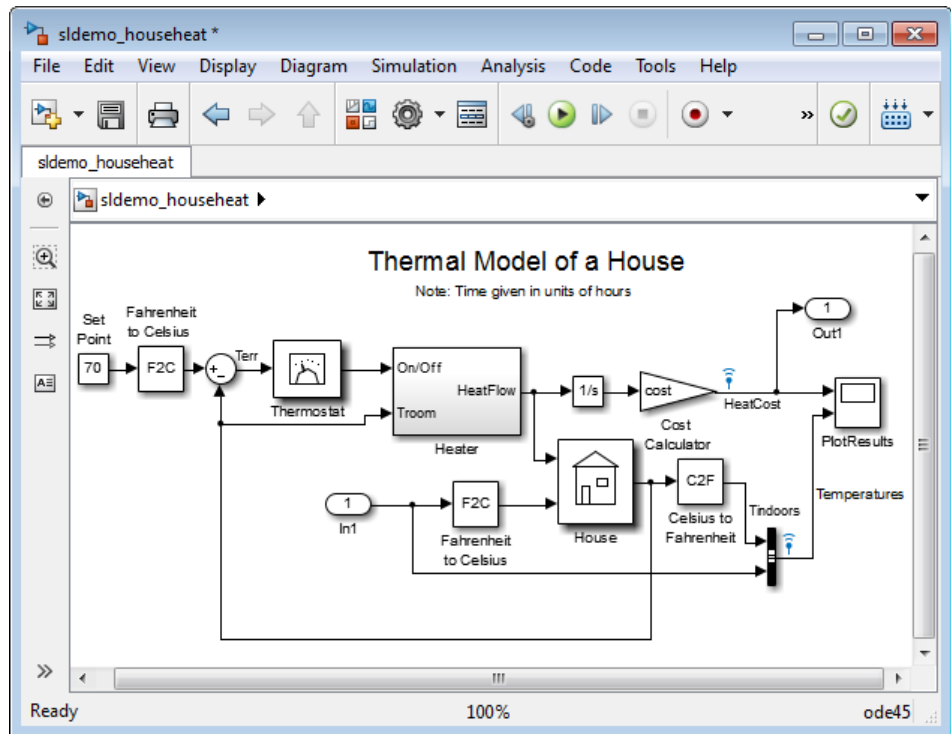
Once you complete a model, you may want to export your simulation results to the MATLAB workspace for further data analysis or visualization. For information about additional data export capabilities, see “Export Runtime Information”.

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

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

An instance of the Out1 block appears in the model window.

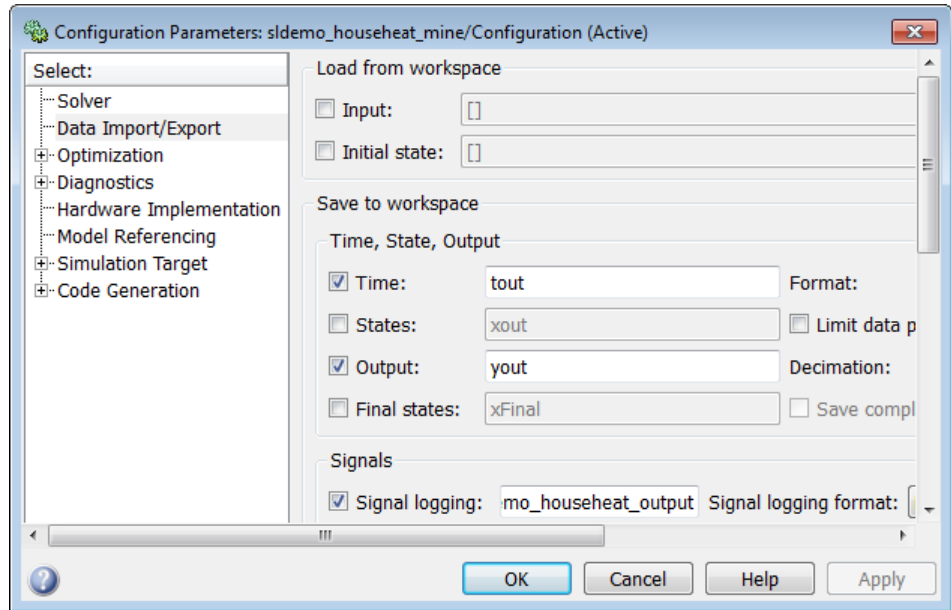
- 3 Draw a *branch line* from the HeatCost signal line to the Out1 block. For more information, see “Draw a Branch Line” on page 3-11.



4 Select **Simulation > Model Configuration Parameters**.

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

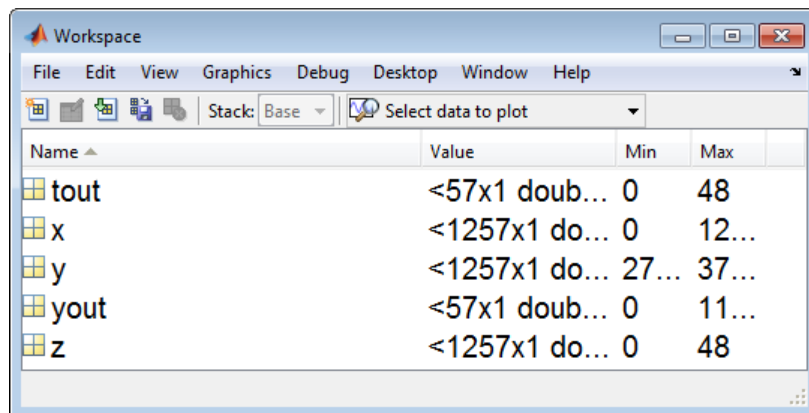
6 On the **Data Import/Export** pane, select the **Time** and **Output** check boxes.



7 Click **OK**. The software applies your changes and closes the dialog box.

8 Rerun the simulation.

The software simulates the model and saves the time and HeatCost data to the MATLAB workspace in the variables `tout` and `yout`.



Basic Simulation Workflow

Simulation with Data Import and Signal Inspection

In this section...

“Data Import and Logging Workflow” on page 5-2

“Example Used in this Tutorial” on page 5-2

“Import Signal Data” on page 5-4

“Simulate Model” on page 5-10

“Compare Signals” on page 5-14

“Create, Save, and Print a Figure” on page 5-16

Data Import and Logging Workflow

Simulink provides two features to import data and view simulation results: the Signal Builder block and the Simulation Data Inspector. Use the Signal Builder block to import input signals into your model. The Simulation Data Inspector allows you to view model output signals and compare them.

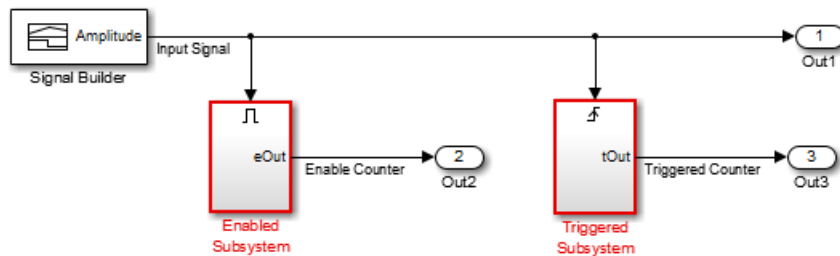
A typical workflow includes the following steps:

- 1** Import signal data from a Microsoft® Excel® spreadsheet into a Signal Builder block. See “Import Signal Data” on page 5-4.
- 2** Simulate the model after enabling the logging of output signals during simulation. See “Simulate Model” on page 5-10.
- 3** View and compare the output signals in the Simulation Data Inspector. See “Compare Signals” on page 5-14.
- 4** Create, print, and save a plot from the Simulation Data Inspector. See “Create, Save, and Print a Figure” on page 5-16.

Example Used in this Tutorial

`sldemo_2counters` is a Simulink model that includes a Signal Builder block and two subsystems. Each subsystem is a counter circuit.

- Enabled Subsystem — Increments the counter when the subsystem is enabled and subsequently retains its output while the subsystem is disabled. Because the Enable port is configured to reset states when enabling, the counter returns to zero when the subsystem is re-enabled.
- Triggered Subsystem — Increments the counter for each rising edge of the control signal.



ex_signaldata.xls is a Microsoft Excel file. It contains two sets of input data on two sheets. The first set of input data is a square wave to test the model response to an impulse. The second set of input data is a triangular wave to test the model response to a ramp signal.

	A	B	C	D	E	F	G	H
1	Time	Amplitude						
2	0.0	0.0						
3	0.2	0.0						
4	0.2	1.0						
5	0.4	1.0						
6	0.4	0.0						
7	0.6	0.0						
8	0.6	1.0						
9	0.8	1.0						
10	0.8	0.0						
11	1.0	0.0						
12	1.0	1.0						
13	1.2	1.0						
14	1.2	0.0						
15	1.4	0.0						
16	1.4	1.0						
17	1.6	1.0						
18	1.6	0.0						
19	1.8	0.0						
20	1.8	1.0						
21	2.0	1.0						
22	2.0	0.0						

Import Signal Data

- “Preparing Signal Builder Block” on page 5-4
- “Importing Data from Excel Spreadsheets” on page 5-6
- “Displaying Input Data” on page 5-8

Preparing Signal Builder Block

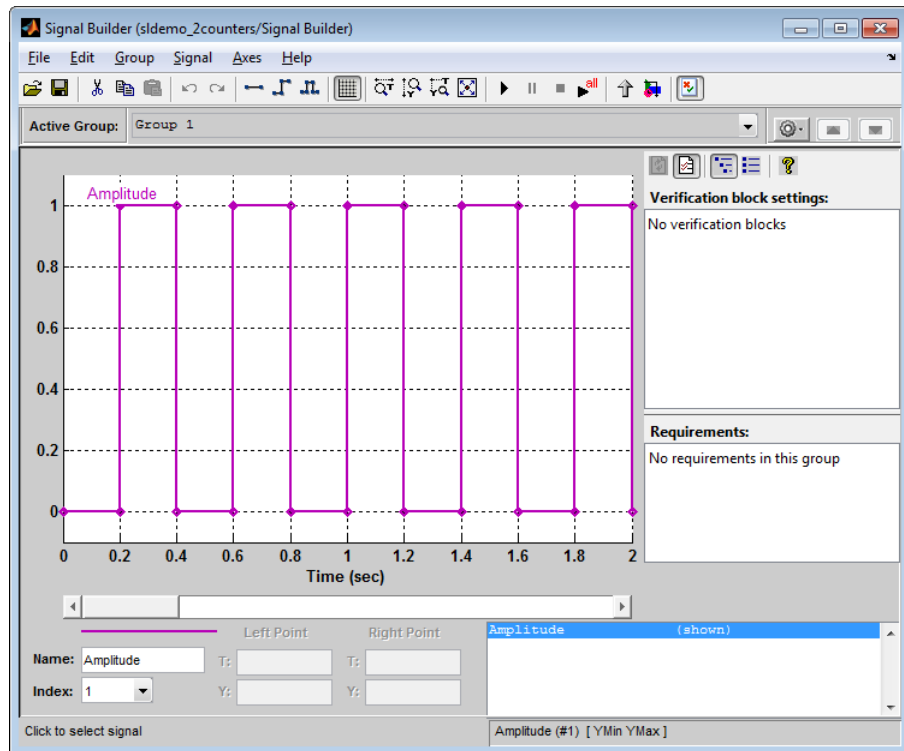
The example Excel spreadsheet contains two sets of input data on two separate sheets. Because the Signal Builder block imports separate sheets into separate groups, you need to create a second group.

- 1 In the MATLAB Command Window, enter:

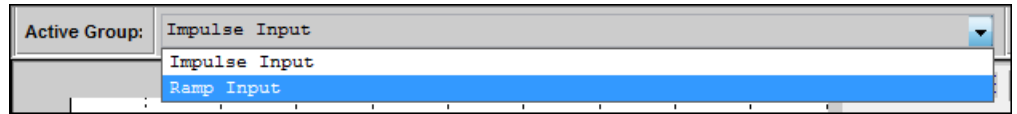
sldemo_2counters

The example model with a Signal Builder block opens in a Simulink editor window.

- 2 Double-click the Signal Builder block. A Signal Builder window opens and displays one data group with an Amplitude signal.



- 3 From the Signal Builder menu, select **Group > Copy**. A second group named **Group 2** appears in the Active Group drop-down list.
- 4 Select **Group > Rename**, and enter more meaningful names that describes the input data. For example, rename Group 1 and Group 2 to **Impulse Input** and **Ramp Input**.



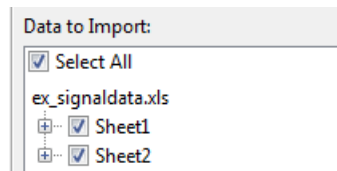
Importing Data from Excel Spreadsheets

Simulink saves data imported from an Excel file using a Signal Builder block with the model and loads the data into memory when you open the model. If you have large sets of data, consider using a MAT-file with the Inport block.

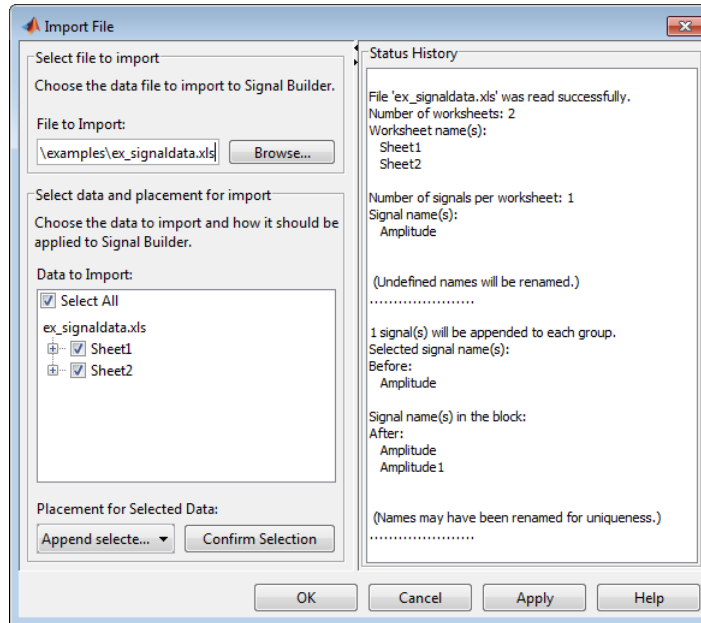
- 1 From the Signal Builder menu, select **File > Import from File**. In the Import File dialog box, click the **Browse** button, find the folder `matlabroot\help\toolbox\simulink\examples`, and select `ex_signaldata.xls`.

The **Data to Import** section of the Import File dialog box displays the Excel file name with a list of sheets in the file.

- 2 Select the **Select All** check box.



- 3 From the **Placement for Selected Data** list, select **Append selected signals to different groups (in order)**. Signal Builder converts the data on each sheet into a group.
- 4 Click the **Confirm Selection** button.

5 Review the **Status History** pane.

The Status History section includes the following information:

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

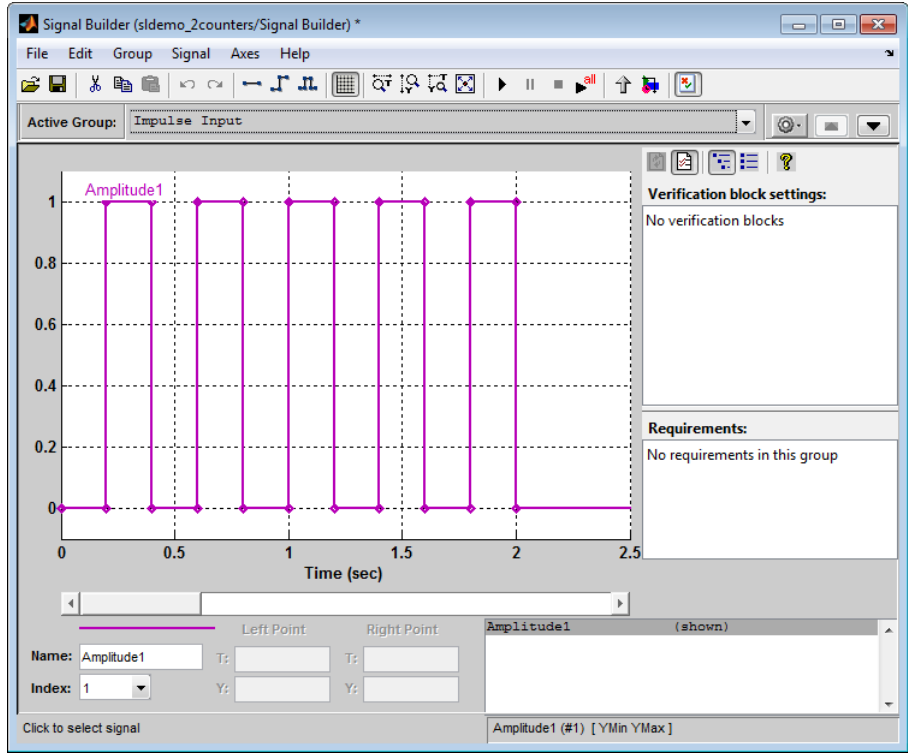
6 Click **OK**.

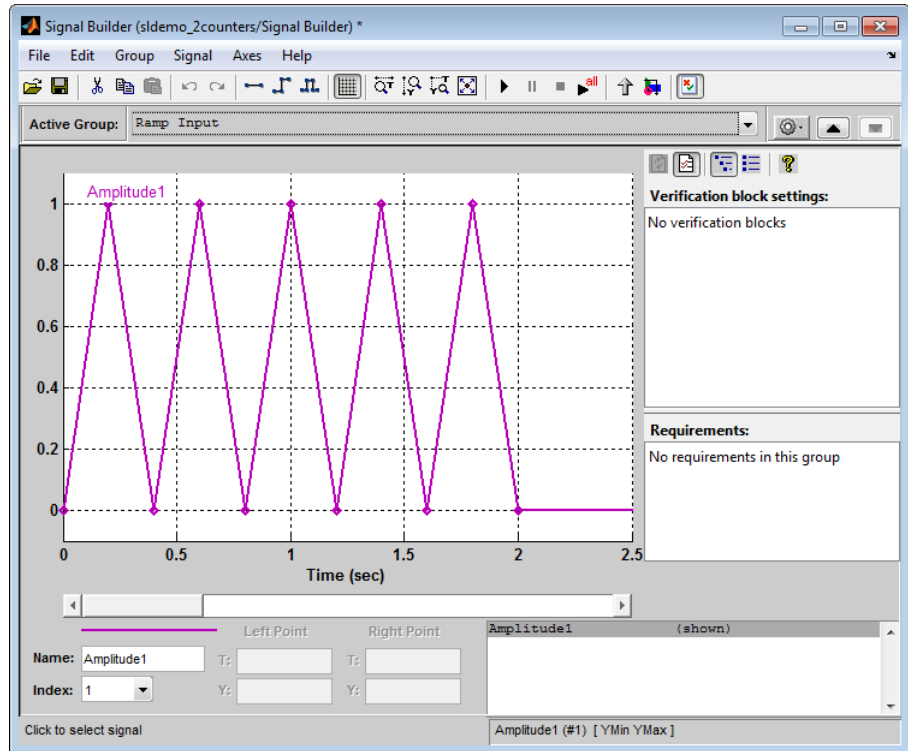
Displaying Input Data

After you import data from the Excel file, you can view the input signals in the Signal Builder window.

- 1 In the Signal Builder window, display the **Ramp Input** group.
- 2 Select the signal labeled Amplitude, and then select **Edit > Delete**. The original Amplitude signals are removed from both the **Impulse Input** and **Ramp Input** groups.
- 3 From the menu, select **Axes > Set T Display Limits**. In the Set time range dialog box, enter **2.5** for the maximum time.
- 4 Repeat Step 3 to set the time range for the other input group.
- 5 Select **File > Save**.

Your signal groups should look like the following. Keep this window open.





If you don't see the wave in either group, click **Amplitude1** in the lower right box.

Note After deleting the original Amplitude signal from the Signal Builder, you may need to reconnect the **Input Signal** line to the **Signal Builder** block in your model.

You can now set up the model to log output signals and simulate. See "Simulate Model" on page 5-10.

Simulate Model

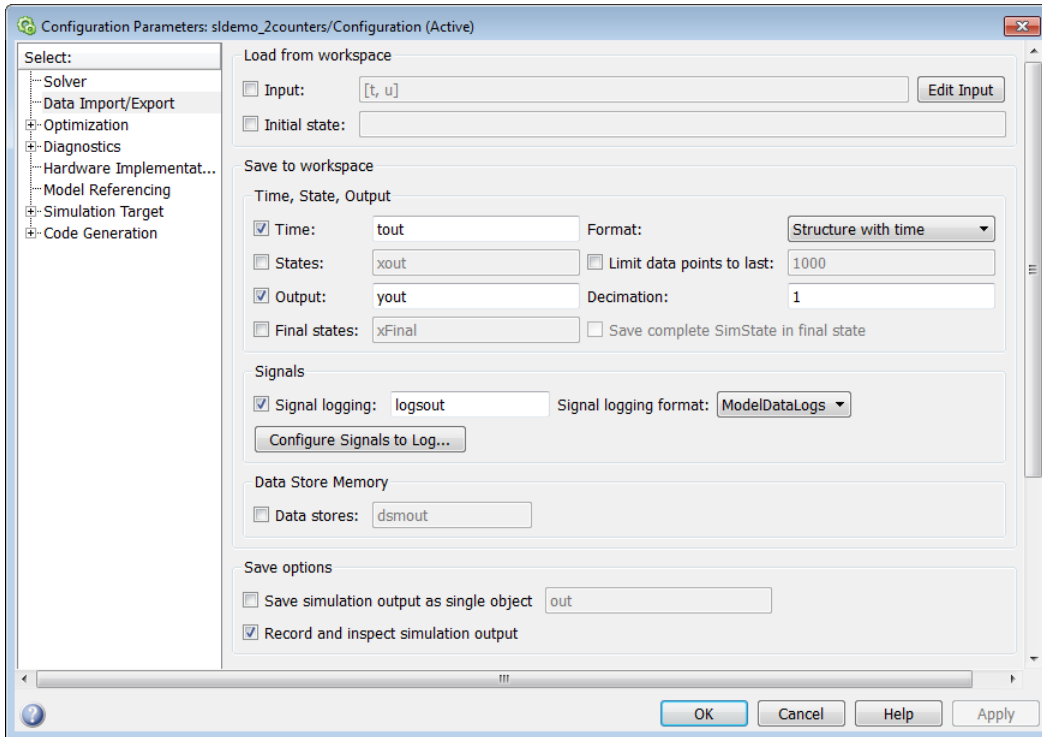
- "Configuring the Model to Log Output Signals" on page 5-11

- “Running the Simulation” on page 5-13

Configuring the Model to Log 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 > Model Configuration Parameters**. The Configuration Parameters dialog box opens to the **Solver** pane.
- 2** In the left pane, select **Data Import/Export**.
- 3** Select the following check boxes:
 - **Time**
 - **Output**
 - **Signal logging**
 - **Record and inspect simulation output**
- 4** From the **Format** list, select **Structure with time**.

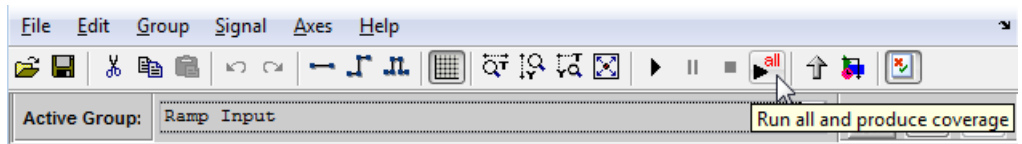


5 Click OK.

Running the Simulation

You can run simulations from the Signal Builder window. Select the input you want to run with this model or run separate simulations with each of the defined inputs.

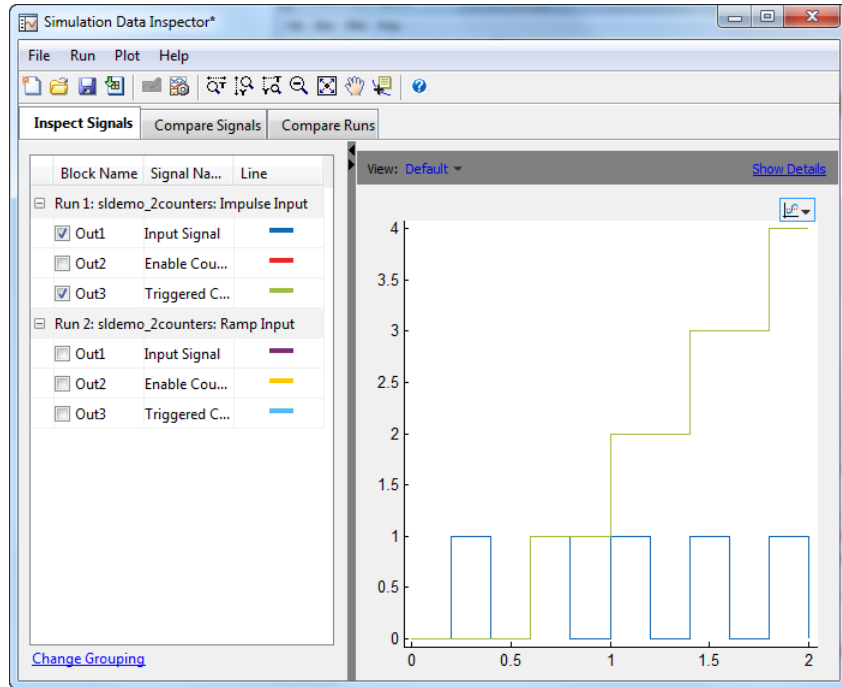
- 1 In the Signal Builder toolbar, click the **Run all** button.



Simulink runs the model simulation from 0.0 s to 2.00 s for the Impulse Input data and then runs again for the Ramp Input amplitude data. The Simulation Data Inspector opens.

- 2 Select the signals that you want to display.

If you select the **Input Signal** and the **Triggered Counter** signal for Run 1, two line plots appear in the right pane of the Simulation Data Inspector window.



Note If you do not see data in the Simulation Data Inspector window, you may need to reconnect the **Input Signal** line to the **Signal Builder** block in your model, and run the simulation again.







Now, you can inspect the results further. See “Compare Signals” on page 5-14.

Compare Signals

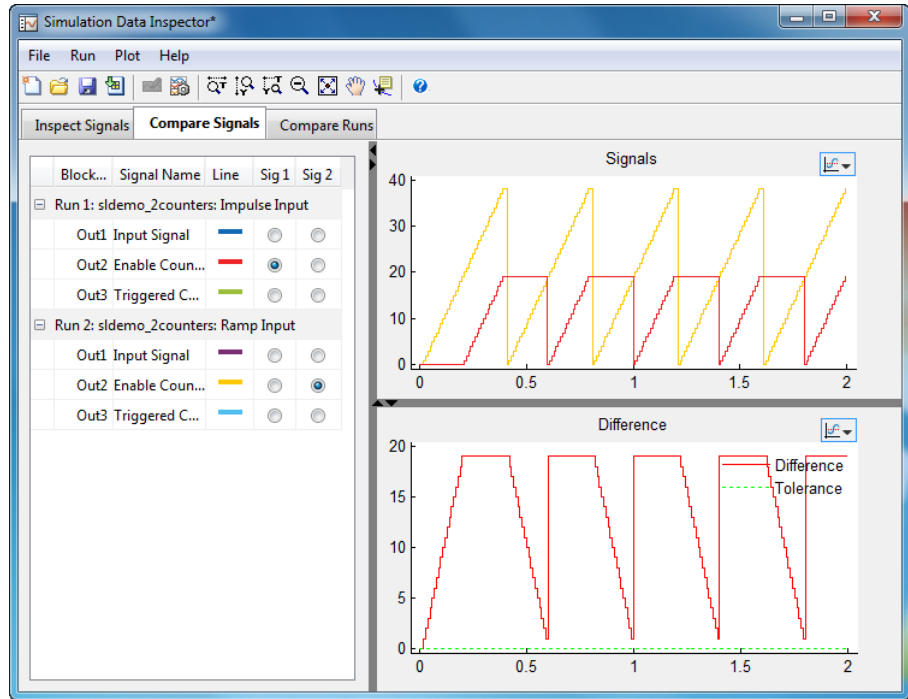
The Simulation Data Inspector allows you to quickly compare two signals from the same or different runs.

- 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. Likewise, in the **Sig 2** column, for Run 2, select the Enable Counter signal.

Inspect Signals		Compare Signals		Compare Runs	
Block Name	Signal Name	Line	Sig 1	Sig 2	
[-] Run 1: sldemo_2counters: Impulse Input					
Out1	Input Signal		<input type="radio"/>	<input type="radio"/>	
Out2	Enable Coun...		<input checked="" type="radio"/>	<input type="radio"/>	
Out3	Triggered Co...		<input type="radio"/>	<input type="radio"/>	
[-] Run 2: sldemo_2counters: Ramp Input					
Out1	Input Signal		<input type="radio"/>	<input type="radio"/>	
Out2	Enable Coun...		<input type="radio"/>	<input checked="" type="radio"/>	
Out3	Triggered Co...		<input type="radio"/>	<input type="radio"/>	

- 3** 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.



To work with your plots as figures, see “Create, Save, and Print a Figure” on page 5-16.

Create, Save, and Print a Figure

- “Creating a Figure from a Plot” on page 5-16
- “Saving and Printing a Figure” on page 5-18

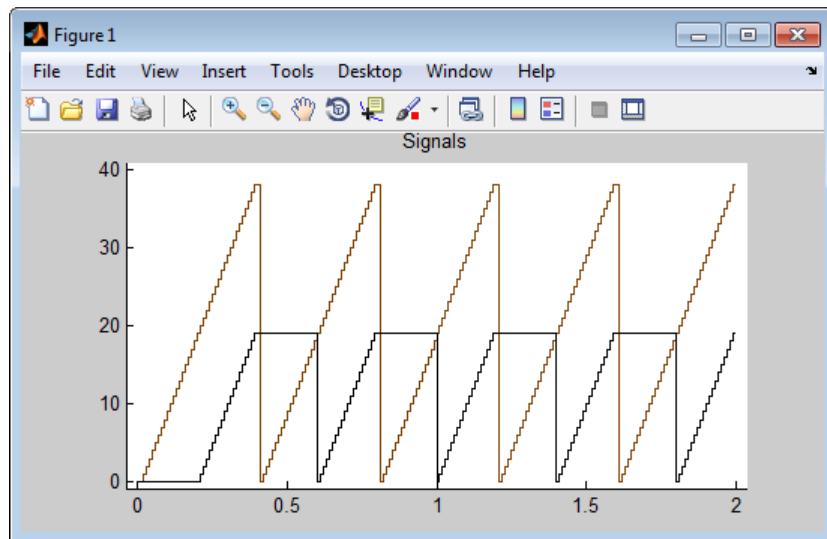
Creating a Figure from a Plot

You can create a figure from your Data Inspector plot, to print or save to your computer.

- 1 In the Simulation Data Inspector window, select the Plot options button in the upper right of a signal plot. From the submenu, select **Plot in new figure**.



A MATLAB figure window opens with the selected signals.



Saving and Printing a Figure

To save your figure to your computer:

- 1** 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 3-8

 copying 3-4

 masking 4-10

 moving 3-7

 ports 3-8

 subsystems 4-6

browser

 library 2-6

C

closing

 model 3-16

Configuration Parameters dialog box 3-13

connecting

 blocks 3-8

 lines to input ports 3-8

control system

 model overview 4-4

 modeling 4-2

copying blocks to a model 3-4

creating

 new model 2-4 3-3

 subsystem mask 4-12

D

data

 exporting to MATLAB 4-25

 importing from MATLAB 4-21

E

editor window

 overview 2-9

example model 3-2

 opening 4-3

 overview 4-4

 running 4-14

examples

 accessing 1-10

exporting

 data from Simulink to MATLAB 4-25

H

help

 Simulink 1-9

house model

 overview 4-4

I

import data

 Excel spreadsheet 5-4

 Signal Builder block 5-4

importing

 MATLAB data to Simulink 4-21

input data

 importing from MATLAB 4-21

input port 3-8

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 3-8
 - connecting to input ports 3-8

M

- masking
 - creating a mask 4-12
 - overview 4-10
- MATLAB
 - exporting data from Simulink 4-25
 - importing data to Simulink 4-21
 - Simulink interaction 1-4
- model configuration
 - logging output 5-11
- model window
 - moving blocks 3-7
- Model-Based Design
 - overview 1-5
 - process 1-6
- modeling
 - control system 4-2
 - process 1-6
- models
 - closing 3-16
 - control system example 4-4
 - creating new 2-4
 - creating simple 3-3
 - empty 3-3
 - opening 2-4
 - opening existing 2-5
 - simple example 3-2
 - simulating 3-13
- moving
 - blocks in the model window 3-7

O

- online help

- accessing 1-9
 - Simulink 1-9
- opening
 - example model 4-3
 - existing model 2-5
 - model 2-4
 - Simulink 2-2
- output data
 - exporting to MATLAB 4-25
- output port 3-8
- overview
 - example model 4-4

R

- running
 - example model 4-14
 - simulation 3-13
 - Simulink 2-2

S

- sample
 - model 3-2
- simulating models
 - exercise 3-13
- simulation
 - running 4-14
 - specifying options 3-13
 - starting 3-14
 - stopping 3-14
- Simulink
 - basics 2-1
 - editor window 2-9
 - examples 1-10
 - getting help 1-9
 - Library Browser 2-6
 - MATLAB interaction 1-4
 - online help 1-9
 - opening 2-2

- starting 2-2
- user interface 2-6
- web resources 1-10
- starting
 - simulation 3-14
 - Simulink 2-2
- stopping
 - simulation 3-14
- subsystems
 - masking 4-10
 - overview 4-6

T

- thermo model

- overview 4-4

U

- user interface
 - Simulink 2-6

W

- web resources
 - Simulink 1-10
- workflow
 - overview 5-2