SIMULINK®

Dynamic System Simulation Software

SIMULINK 2
November 1996

New Features

How to Contact The MathWorks:

508-647-7000 Phone

508-647-7001 Fax

The MathWorks, Inc.
24 Prime Park Way
Natick, MA 01760-1500 Mail

`http://www.mathworks.com` Web
`ftp.mathworks.com` Anonymous FTP server
`comp.soft-sys.matlab` Newsgroup

`support@mathworks.com` Technical support
`suggest@mathworks.com` Product enhancement suggestions
`bugs@mathworks.com` Bug reports
`doc@mathworks.com` Documentation error reports
`subscribe@mathworks.com` Subscribing user registration
`service@mathworks.com` Order status, license renewals, passcodes
`info@mathworks.com` Sales, pricing, and general information

SIMULINK New Features (November 1996)

Printing History: Jan 1996 First printing
Nov 1996 Third printing

Contents

Introduction

This document provides a brief description of many of the significant features to be included in SIMULINK 2. The organization of the manual reflects the order of topics in *Using SIMULINK*.

Loading SIMULINK 1.3 Models

SIMULINK 2 loads SIMULINK 1.3 models without any problems, although a warning message indicates that you are loading a model created by a previous version of SIMULINK. SIMULINK automatically converts older models to the SIMULINK 2 model file format and marks a loaded model as modified; when you close a model, SIMULINK asks whether you want to save the modified file.

Updating SIMULINK 1.3 Models

The new `slupdate` command updates SIMULINK 1.3 models containing specific blocks to SIMULINK 2 format. This command updates these blocks:

- The Pulse Generator block has been rewritten.
- The Hit Crossing block is now a built-in block.
- The Memory block is now a built-in block.
- The Quantizer block is now a built-in block.
- The Graph scope's function has been replaced with the new Scope block.
- The 2-D Table Look-Up block is now a built-in block and has been renamed Table Look-Up (2-D).
- The Limited Integrator block's function is now part of the Integrator block.

Also, `slupdate` calls `addterms` to terminate any unconnected input and output ports by attaching Ground and Terminator blocks, respectively. These blocks are described in *Using SIMULINK*.

To update a SIMULINK 1.3 model, open the model, then enter

```
slupdate('sys')
```

where `sys` is the model name.

For each out of date block, SIMULINK asks whether it should be updated.

Loading Models Containing Reset Integrator Blocks

Although the function performed by the Reset Integrator block has been built into the Integrator block, `slupdate` does not replace Reset Integrator blocks with Integrator blocks. It is difficult to correctly configure the new Integrator block to satisfy all the different ways the Reset Integrator block could be used in SIMULINK 1.3 models. The Reset Integrator block is not available in the Linear library; you are encouraged to apply the Integrator block instead.

Changes to the SIMULINK User Interface

SIMULINK 2 has a substantially new look, with a new menu structure, redesigned block dialog boxes, and revised block library layout.

All SIMULINK commands are organized under four top-level menus: **File**, **Edit**, **Simulation**, and **Format**. The menu items are described in the sections of the manual that discuss the functions they perform.

The **Simulation Parameters** dialog box is new for SIMULINK 2. Changes are discussed in "Running a Simulation" on page 6.

The **Mask Editor** dialog box has been redesigned, making it easier to mask blocks. Changes to masking are described in "Using Masks to Customize Blocks" on page 8.

Improved Documentation

Using SIMULINK has been completely rewritten and has been improved in many ways:

- The new manual combines the previous edition of the *SIMULINK User's Guide* and the *SIMULINK Version 1.3 Release Notes* into a single book.
- An expanded chapter on building models provides clearer and more complete instructions. A useful table of keyboard shortcuts offers a quick guide for more experienced users.
- Additional chapters and appendices provide information not previously available.

Creating a Model

For more information about the features discussed in this section, see Chapter 3 of *Using SIMULINK*.

Starting SIMULINK

You can start SIMULINK 2 in these ways:

- Enter a model file name to display the block diagram for that model (the same way you did using SIMULINK 1.3).
- Enter `simulink` in the MATLAB command window to display the SIMULINK block library.
- Windows and Macintosh users can click on the SIMULINK toolbar button to display the SIMULINK block library and get a new model window. The toolbar button looks like this:

Macintosh users can launch MATLAB by double-clicking on a model file icon.

Model and Block Parameters

Many model and block parameter names have changed in SIMULINK 2. If you use `set_param` commands to set model or block parameters, you need to make sure the parameter names are correct. Chapter 11 of *Using SIMULINK* discusses the `set_param` command and Appendix E lists model and block parameters.

Working with Blocks

SIMULINK 2 provides many enhancements for blocks and block libraries.

Changes to the Block Libraries

The SIMULINK block libraries have been modified. Each library has a new icon that depicts the kind of blocks it contains. Also, the layout of block icons within each library has been modified to make it easier to find the block you want. Many block icons have been changed and icon sizes are more consistent.

The Extras Library

The Extras block library, which contains blocks that supplement built-in blocks, has been moved to the new Blocksets and Toolboxes library. When you open that icon, SIMULINK searches through your installed software and displays icons for any blocksets and toolboxes it finds. The Extras library always appears in the list. To examine blocks in the Extras library, open that icon. Blocks are organized into these sublibraries: Additional Sinks, Additional Discrete, Additional Linear, Coordinate Conversions, Flip Flops, and Linearization.

The Demos Library

A separate and expanded Demos library provides more demo models that illustrate SIMULINK 2 features.

Changes to Block Dialog Boxes

All block dialog boxes have been redesigned. Block descriptions (these appear below the block type on the dialog box) have been rewritten to provide more useful information. The **Help** button now accesses the reference page for the block, using the new SIMULINK Block Browser.

Moving a Block Name

You can move a block name to the opposite side of the block by dragging the name. This is equivalent to using the **Flip Name** menu item from the **Format** menu.

Additional Colors

SIMULINK 2 provides more colors for the screen background and the block background and foreground. The new colors are Light Blue, Dark Green, Orange, and Gray.

Working with Lines

SIMULINK 2 provides many enhancements for connecting lines.

Drawing Lines Between Blocks

SIMULINK 2 draws connecting lines either as straight lines or as perpendicular horizontal and vertical line segments. If you hold down the **Shift** key while drawing a connecting line, SIMULINK draws a straight diagonal line that

snaps to the target input port. When you draw a line to connect blocks, if the cursor is within the target block, the line is connected to the closest input port.

Moving Line Segments

In SIMULINK 2, you can move a horizontal or vertical line segment in two ways. You can move the segment parallel to its original position, or you can move the segment and the segments connected to it.

Labeling Signals

In SIMULINK 2, you can label signals to annotate your model. Signal labels can be located at either end or at the center of one or more line segments. Labels remain attached to lines as they are moved. Labels themselves can be copied or moved using drag-and-drop techniques.

Signal Label Propagation

SIMULINK 2 supports the propagation of signal labels through blocks in the Connections block library. Signal label propagation is the automatic labeling of a line carrying a signal that is labeled someplace else in a model. The labeled signal must be on a line feeding a connecting block and the propagated signal must be on a line coming from the same connecting block or one associated with the first.

Labeling Ports

SIMULINK 2 labels ports on Subsystem blocks using the Inport and Outport block names in the underlying subsystem.

Closing a Model

When a model is closed, its `CloseFcn` callback is called.

Copying a Model to the PC Clipboard

Windows users can copy the block diagram into the clipboard for use with another application. Choose **Copy Model** from the **Edit** menu. The default copy format is Windows Metafile. You can change the format from the MATLAB command window by choosing **Preferences** from the **File** menu. Then, select the **Copying Options** tab.

Running a Simulation

SIMULINK 2 provides many improvements in this area. All features discussed in this section are described in detail in Chapter 4 of *Using SIMULINK*.

Expanded Simulation Parameters Dialog Box

A new **Simulation Parameters** dialog box provides more information about solvers and gives you more error control. Also, it is now easier to manage workspace I/O and levels of diagnostic messages and intervention. The **Simulation Parameters** dialog box consists of three "pages." The **Solver** page enables you to select a solver and specify its parameters. The **Workspace I/O** page enables you to manage input from and output to the workspace. The **Diagnostics** page enables you to control the level of intervention for certain events.

In SIMULINK 1.3, to provide output at specified times, you would use the `HitTimes` parameter. In SIMULINK 2, you select the **Produce additional output** choice on the **Output options** list, on the **Solver** page.

Running a Simulation from the Command Line

In SIMULINK 2 you can use the `sim` and `simset` commands to run a simulation from the command line. These commands give you more control of simulation parameters and provide command line access to all parameters that can be set on the **Simulation Parameters** dialog box. The `simget` command enables you to obtain values of simulation parameters and solver properties for a model.

If you access a model using the block diagram S-function API, you can specify flag values using special strings.

State-of-the-Art Integrators

SIMULINK 2 incorporates the set of integration algorithms developed for the MATLAB ODE suite. These solvers provide faster, more accurate simulation results. The ODE suite includes variable-order and fixed-step nonstiff and stiff solvers. Selecting the appropriate solver is easier with the improved **Simulation Parameters** dialog box.

An important additional benefit of the new solvers is that it is no longer necessary, or even advisable, to adjust step size to get better granularity in the simulation results. The variable-step solvers automatically set step sizes to

provide accurate results. Also, because SIMULINK 2 provides fixed-step solvers, it is no longer either necessary or desirable to modify step sizes.

This table indicates, for each integration method supported in SIMULINK 1.3, the corresponding solver provided in SIMULINK 2.

Table 1: SIMULINK 1.3 Integrators and SIMULINK 2 Solvers

If you used this integrator in SIMULINK 1.3	Consider using this solver in SIMULINK 2
`linsim`	`ode45` (nonstiff) or `ode15s` (stiff)
`rk23`	`ode23`
`rk45`	`ode45`
`adams`	`ode113`
`gear`	`ode15s`
`euler`	`ode1`

Zero Crossing Detection

A *zero crossing* occurs when a signal makes a transition to zero, crosses zero, or makes a transition from zero. A zero crossing also occurs when a signal reaches some defined threshold (not necessarily at zero), such as an upper limit in a Saturation block.

Many SIMULINK blocks have built-in detection of zero crossings. SIMULINK can detect the point where a signal crosses zero to within computer tolerance. This feature, which results in more accurate simulations, is discussed in Chapter 10 of *Using SIMULINK*.

You can disable zero crossing detection for a simulation. On the **Simulation Parameters** dialog box, select the **Diagnostics** tab. Then, in the Debugging area, select the **Disable zero crossing detection** check box.

Using Masks to Customize Blocks

Masking enables users to create block dialog boxes or customize block icons. Masking is described in Chapter 6 of *Using SIMULINK*.

SIMULINK 2 provides these enhancements to masking:

- A new user interface
- An easier way to define dialog box prompts, including a more direct way to associate a variable to a block parameter
- The ability to add pop-up menus and check boxes to the mask dialog box
- An improved way to define initialization commands, including the ability to assign a user-entered value to a variable without evaluating it
- A separate workspace for each masked block (similar to an M-file function)
- A simpler way to examine a block's mask
- The ability to examine the blocks in a masked subsystem without destroying the mask
- New plotting options
- New options for controlling the appearance of icons

In SIMULINK 2, mask parameter names have changed. The integrity of masked blocks created using all prior SIMULINK versions is preserved. Mask parameters are listed in Appendix E of *Using SIMULINK*.

Conditionally Executed Subsystems

Conditionally executed subsystems are subsystems that may or may not execute, depending on their input. Conditionally executed subsystems are described in Chapter 7 of *Using SIMULINK*.

SIMULINK 2 provides support for three types of conditionally executed subsystems:

- An *enabled subsystem* executes while the control signal is positive, starting execution at the simulation step where the control signal crosses zero (from the negative to the positive direction) and continuing execution while the control signal remains positive.
- A *triggered subsystem* executes at the simulation step when a trigger event occurs. A trigger event can occur on the rising or falling edge of a trigger signal, which can be continuous or discrete.
- A *triggered and enabled subsystem* executes once on the simulation step when a trigger event occurs if the enable control signal has a positive value at that step.

Conditionally executed subsystems are useful in a variety of applications. For example, in the automotive industry, triggered subsystems can be used to model the dynamics of an internal combustion engine. In the aerospace industry, enabled subsystems can be used to model complex flight control laws, where different controllers are enabled during different flight regimes.

SIMULINK Blocks

This section discusses new, revised, and obsoleted blocks. All blocks are described in Chapter 9 of *Using SIMULINK*.

Online Block Reference Browser

The current reference pages for SIMULINK blocks are available as online help. You can access these pages in two ways:

- By clicking on the **Help** button on any block dialog box. The reference page for the block is displayed.
- By accessing the MATLAB Help Desk, then selecting **SIMULINK Blocks** from the SIMULINK Topics area.

  If you're using a PC or a Macintosh, you access the MATLAB Help Desk by clicking on the **Help** toolbar button (a question mark), selecting the **Help Desk** menu item from the MATLAB **Help** menu, or entering `helpdesk` in the command window.

  If you're using a UNIX computer, you access the MATLAB Help Desk by entering the `helpdesk` command in the MATLAB command window.

Revised Block Dialog Boxes

All block dialog boxes have been redesigned for SIMULINK 2. In addition to a more visually appealing layout, each dialog box includes an **Apply** button to accept current settings and keep the dialog box open, a **Revert** button to restore the original settings when the block was most recently opened or since the **Apply** button was last pressed, a **Close** button that applies the changes and closes the dialog box, and a **Help** button that accesses the Online Block Reference Browser, described in the previous section.

The block descriptions have been rewritten to provide more helpful and consistent information.

Vectorization of Blocks

A *vectorized* block can accept a vector input signal or generate a vector output signal, or both. In SIMULINK 2, almost all blocks are vectorized. To find out whether a block is vectorized, consult the online Block Browser or check the reference page for the block in the manual. At the end of each block reference

page a table provides information about block characteristics, including whether the block is vectorized.

New and Enhanced Blocks

SIMULINK 2 provides several new and enhanced blocks. All blocks are described in detail in Chapter 9 of *Using SIMULINK* and in the Online Block Browser.

Algebraic Constraint

The Algebraic Constraint block constrains the input signal f(*z*) to zero and outputs an algebraic state *z*. The block outputs the value necessary to produce a zero at the input. The output must affect the input through some feedback path. This enables you to specify algebraic equations for index 1 differential/algebraic systems (DAE's).

Backlash

The Backlash block no longer has an **Initial input** parameter. The initial center of the deadband width is defined by the **Initial output** parameter.

Data Store Memory, Data Store Read, and Data Store Write

The Data Store Memory, Data Store Read, and Data Store Write blocks enable the model to write and read data to and from a memory region during a simulation.

Discrete Filter

The Filter block has been renamed Discrete Filter.

Discrete-Time Integrator

The Discrete-Time Integrator block enables you to define limits on the integration, which provides the capabilities of the (removed) Discrete-Time Limited Integrator.

Display

The Display block shows the value of its input signal. You can control the display format and the frequency of the display.

Dot Product

The Inner Product block has been renamed Dot Product.

Elementary Math

The Elementary Math block performs numerous common mathematical functions. The block output is the result of applying the selected function to the input.

Enable

The Enable block is used with conditionally executed subsystems. Adding an Enable block to a subsystem creates an *enabled* subsystem. Adding both an Enable and a Trigger block creates an *triggered and enabled* subsystem. For more information about conditionally executed subsystems, see page 9 of *SIMULINK New Features* or Chapter 7 of *Using SIMULINK*.

Fcn

The rules of precedence for operations for the Fcn block did not conform to the industry standard. The rules implemented in the current block do.

Filter

The Filter block has been renamed Discrete Filter.

From

The From block, when used with a Goto block, provides a convenient way to pass a signal from one block to another without physically connecting the blocks.

Goto

The Goto block, when used with a From block, provides a convenient way to pass a signal from one block to another without physically connecting the blocks.

Goto Tag Visibility

The Goto Tag Visibility block defines the scope of a Goto block tag.

Ground

Connecting a Ground block to a block's input port prevents SIMULINK from issuing a warning message about the block's unconnected port. The block outputs a zero valued signal.

Hit Crossing

A rewritten Hit Crossing block accurately enables you to detect when the input signal crosses a particular value.

IC

The IC block enables you to define an initial value for a signal.

Inner Product

The Inner Product block has been renamed Dot Product.

Integrator

The revised Integrator block now combines features included in the Limited Integrator and Reset Integrator blocks. In addition, you can add a port on the block to output the state. Also, you can specify the absolute tolerance for the block's state.

Memory

You can choose whether or not the Memory block's sample time is inherited from its driving block. The block dialog box contains a check box labeled **Inherit sample time**. If the check box is selected, the block inherits its sample time from the driving block. If the box is not selected, the block's sample time is continuous.

All Memory blocks in existing models have a sample time of continuous although the default for Memory blocks copied from the Nonlinear library have a sample time of inherited. For Memory blocks to work as they did in SIMULINK 1.3, make sure the check box is not selected.

MinMax

The MinMax block detects either the minimum or maximum of its input signal(s).

Multiport Switch

The Multiport Switch block chooses a block input from among multiple inputs. An integer-valued control input determines which input to pass through to the output port.

Note

The Note block has been removed. You can provide model annotations by creating an annotation, described in Chapter 3 of *Using SIMULINK*.

Outport

The Outport block adds an option that, when used in a conditionally executed subsystem, allows the block to control whether its output is reset to an initial value or held at its most recent value while the subsystem is disabled.

Pulse Generator

The Pulse Generator block parameters have changed: The **Pulse period** parameter is now **Period**, the **Pulse width** parameter is now **Duty cycle**, the **Pulse height** parameter is now **Amplitude**, and the **Pulse start time** parameter is now **Start time**.

Ramp

The Ramp block provides a signal that starts at a specified time and value and changes by a specified rate.

Rate Limiter

The **Rising slew rate** and **Falling slew rate** parameters now accept values of `inf` and `-inf`, respectively. These values pass the input through the block without applying limits.

Relay

The Relay block **Input for on** and **Input for off** parameters have been renamed to **Switch on point** and **Switch off point**.

Scope

An enhanced oscilloscope-like Scope block provides vastly improved graphical display of signals. The Scope block allows you to zoom in on the block input in the *x* (time) direction, *y* direction, or both directions; display all the input to the

block; limit the data displayed; and save the signal data to the workspace at the end of simulation.

Selector

The Selector block acts like a “patch panel” for cross wiring of input vector elements. You enter a vector parameter that indicates the input vector elements that make up the block output.

Signal Generator

The Signal Generator’s dialog box has been rearranged to simplify choosing a wave form and defining signal parameters. The default frequency is now Hertz.

Step

The Step Fcn block has been renamed Step.

Terminator

Connecting a block’s output port to a Terminator block prevents SIMULINK from issuing warning messages about unconnected ports. The block does not process the signal.

To File

The To File block provides **Decimation** and **Sample time** parameters to limit the amount of data written to the file.

To Workspace

The To Workspace block provides **Decimation** and **Sample time** parameters to limit the amount of data written to the workspace variable.

Trigger

The Trigger block is used with conditionally executed subsystems. Adding a Trigger block to a subsystem creates a *triggered* subsystem. Adding both and Enable and a Trigger block creates a *triggered and enabled* subsystem.

2-D Look-Up Table

The 2-D Look-Up Table has been renamed Look-Up Table (2-D).

Variable Transport Delay

In SIMULINK 1.3, the **Initial input** parameter was not being set. This has been corrected in the current release.

Width

The Width block generates as output the width of the input vector.

Removed Blocks

Several blocks are no longer available. The functions they perform are included in other blocks:

- The Discrete-Time Limited Integrator block has been replaced by the Discrete-Time Integrator block.
- The Limited Integrator block has been replaced by the Integrator block.
- The Reset Integrator block has been replaced by the Integrator block.

S-Functions

If you are writing C MEX-file S-functions and place an S-function in an enabled subsystem configured to reset its states, the `mdlInitializeConditions` function is called upon reset. To figure out if `mdlInitializeConditions` is called from a reset or at simulation start, you can use the `ssIsFirstInitCond(S)` macro.

Additional Topics

Algebraic Loops

The algebraic loop solver has been improved for SIMULINK 2 and is able to solve a larger class of algebraic loops. It can now attempt to solve algebraic loops that have multirate components, as well as loops containing blocks with nonsmooth outputs (such as the Abs, Saturation, or Quantizer blocks).

Algebraic loops and the algebraic loop solver are described in Chapter 10 of *Using SIMULINK*.

Model Construction Commands

Several model construction commands have been added:

- `find_system`
- `gcb`
- `gcs`
- `get_param`
- `new_system`
- `open_system`
- `save_system`

These commands are described in Chapter 11 of *Using SIMULINK*.

Also, the new `Lines` property enables you to obtain a structure array of all the lines in a block diagram:

```
get_param('sys','Lines')
```

returns a structure array of all the lines in the model named `sys`. The form of each structure in the array is:

- `Handle` is the handle to the line
- `Name` is the line's name
- `Parent` is the handle to the subsystem or block diagram owning the line
- `SrcBlock` is the handle to the source block driving the line
- `SrcPort` is the port number of the source block driving the line
- `DstBlock` is the handle to the block being driven by the line
- `DstPort` is the port number of the destination block
- `Points` is the array of points describing the line
- `Branch` is the structure array of any branch lines on this line

Model File Format

SIMULINK 2 saves models in a structured file format, which results in faster loading and saving of models, and produces more readable model files. Also, model files are now easier to post-process or transfer to other applications.

The model file format is described in Appendix C of *Using SIMULINK*.