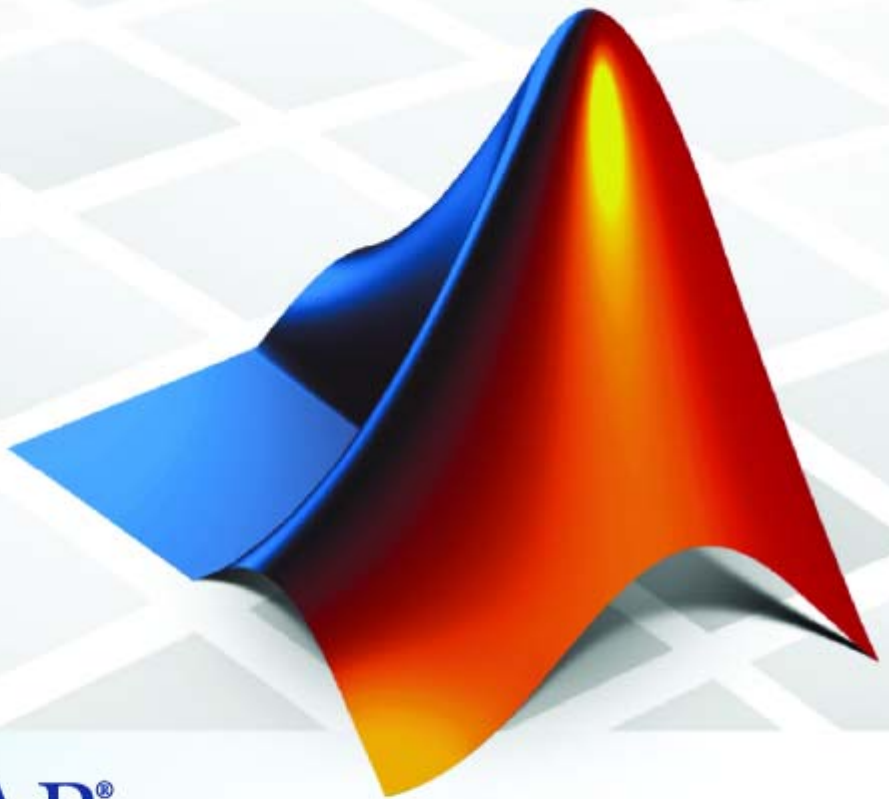


Simscape 1

User's Guide



MATLAB[®]
& **SIMULINK[®]**

How to Contact The MathWorks



www.mathworks.com
[comp.soft-sys.matlab](mailto:comp.soft-sys.matlab@mathworks.com)
www.mathworks.com/contact_TS.html

Web
Newsgroup
Technical Support



suggest@mathworks.com
bugs@mathworks.com
doc@mathworks.com
service@mathworks.com
info@mathworks.com

Product enhancement suggestions
Bug reports
Documentation error reports
Order status, license renewals, passcodes
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.

Simscape User's Guide

© COPYRIGHT 2007 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, Simulink, Stateflow, Handle Graphics, Real-Time Workshop, and xPC TargetBox are registered trademarks, and SimBiology, SimEvents, and SimHydraulics are trademarks of The MathWorks, Inc.

Other product or brand names are trademarks or registered trademarks of their respective holders.

Patents

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

Revision History

March 2007 Online only New for Version 1.0 (Release 2007a)

Getting Started

1

What Is Simscape?	1-2
Simscape and Physical Modeling	1-2
Getting Online Help for Simscape	1-2
Related Products	1-4
Requirements for Simscape	1-4
Other Related Products	1-4
Basic Principles of Modeling Physical Networks	1-5
Overview of the Physical Network Approach to Modeling	
Physical Systems	1-5
Variable Types	1-7
Building the Mathematical Model	1-8
Direction of Variables	1-9
Connector Ports and Connection Lines	1-11
Connecting Simscape Diagrams to Simulink Sources and	
Scopes	1-13
Working with Physical Units	1-14
Unit Definitions	1-14
Specifying Units in Block Dialogs	1-17

Modeling Physical Systems

2

Introducing the Simscape Block Libraries	2-2
Library Structure Overview	2-2
Using the Simulink Library Browser to Access the Block	
Libraries	2-2
Using the Command Prompt to Access the Block	
Libraries	2-3

Essential Steps to Building a Physical Model	2-5
Creating a Simple Model	2-7
Building a Simscape Diagram	2-7
Modifying Initial Settings	2-15
Running the Simulation	2-16
Adjusting the Parameters	2-19
Modeling Best Practices	2-25
Example of Using a Parasitic Resistance to Avoid Numerical Simulation Issues	2-26

Simulating Physical Models

3

How Simscape Works	3-2
Model Validation	3-3
Network Construction	3-4
Equation Construction	3-4
Computing Initial Conditions	3-5
Performing Transient Initialization	3-6
Transient Solve	3-6
Linearizing Simscape Models	3-7
Limitations	3-9
Solver Support	3-9
Signal Support in the Simulink-PS Converter Block	3-9
Algebraic Loops	3-9
Code Generation	3-10

Using the Simscape Editing Mode

4

About the Simscape Editing Mode	4-2
--	------------

Suggested Workflows	4-2
What You Can Do in Restricted Mode	4-3
What You Can Do in Full Mode	4-3
Switching Between Modes	4-4
Working with Block Libraries	4-7
Working with Restricted and Full Modes	4-9
Setting the Model Loading Preference	4-9
Saving a Model in Restricted Mode	4-10
Working with a Model in Restricted Mode	4-12
Switching from Restricted to Full Mode	4-20
Editing Mode Information	4-23
What Is the Current Mode?	4-23
Which Licenses Are Checked Out?	4-23

Glossary

Examples

A

Getting Started	A-2
Best Practices	A-2
Editing Mode	A-2

Index

Getting Started

What Is Simscape? (p. 1-2)

Related Products (p. 1-4)

Basic Principles of Modeling Physical
Networks (p. 1-5)

Working with Physical Units
(p. 1-14)

Overview of the product

Other products you need or might
want to use with Simscape

Overview of modeling techniques

Units and unit conversion functions
available with Simscape

What Is Simscape?

Simscape is a modeling environment for the engineering design and simulation of physical systems within Simulink® and MATLAB®. It serves as a platform for a group of products in the Simulink Physical Modeling family that are all based on physical connections between their blocks. Simscape contains functions and utilities that support functionality common for these products. Simscape also extends Simulink by including the Physical Network approach to modeling physical systems. It contains a comprehensive library of electrical, hydraulic, and one-dimensional translational and rotational mechanical elements and utility blocks.

Simscape and Physical Modeling

Simscape is the platform product for a group of products in the Simulink Physical Modeling family, encompassing the modeling and design of systems according to basic physical principles. Simscape runs within the Simulink environment and interfaces seamlessly with the rest of Simulink and with MATLAB. Unlike other Simulink blocks, which represent mathematical operations or operate on signals, Simscape blocks represent physical components or relationships directly.

Note This Simscape User's Guide assumes that you have some experience with modeling physical systems, as well as with building and running models in Simulink.

Getting Online Help for Simscape

You can get help online in a number of ways to assist you while you use Simscape.

Using the MATLAB Help System for Documentation and Demos

The MATLAB Help browser allows you to access the documentation and demo models for all the MATLAB and Simulink based products that you have installed. The online Help includes an online index and search system.

Consult the “Help Browser” section of the Getting Started with MATLAB documentation for more about the MATLAB Help system.

Related Products

Simscape is the platform product for the Physical Modeling product family, expanding Simulink capabilities to model and simulate physical systems.

Requirements for Simscape

You must have the following products installed to use Simscape:

- MATLAB
- Simulink

Other Related Products

The related products listed in the Simscape product page at the MathWorks Web site include toolboxes and blocksets that extend the capabilities of MATLAB and Simulink. These products will enhance your use of Simscape in various applications.

The Physical Modeling Product Family

Simscape is the platform product for the following vertical products in the Physical Modeling product family:

- SimHydraulics™, for modeling and simulating hydraulic systems
- SimDriveline, for modeling and simulating powertrain systems
- SimMechanics, for modeling and simulating general mechanical systems

The Physical Modeling product family also contains SimPowerSystems, for modeling and simulating electrical power systems. Use these products together to model physical systems in Simulink.

For Information About MathWorks Products

For more information about any MathWorks software products, see either

- The online documentation for that product if it is installed
- The MathWorks Web site at www.mathworks.com

Basic Principles of Modeling Physical Networks

Simscape is a set of block libraries and special simulation features for modeling physical systems in the Simulink environment. It extends Simulink with the Physical Network approach, which differs from the standard Simulink modeling approach and is particularly suited to simulating systems that consist of real physical components.

This section covers the following topics:

- “Overview of the Physical Network Approach to Modeling Physical Systems” on page 1-5
- “Variable Types” on page 1-7
- “Building the Mathematical Model” on page 1-8
- “Direction of Variables ” on page 1-9
- “Connector Ports and Connection Lines” on page 1-11
- “Connecting Simscape Diagrams to Simulink Sources and Scopes” on page 1-13

Overview of the Physical Network Approach to Modeling Physical Systems

Simulink blocks represent basic mathematical operations. When you connect Simulink blocks together, the resulting diagram is equivalent to the mathematical model, or representation, of the system under design. Simscape lets you create a network representation of the system under design, based on the Physical Network approach. According to this approach, each system is represented as consisting of functional elements that interact with each other by exchanging energy through their ports.

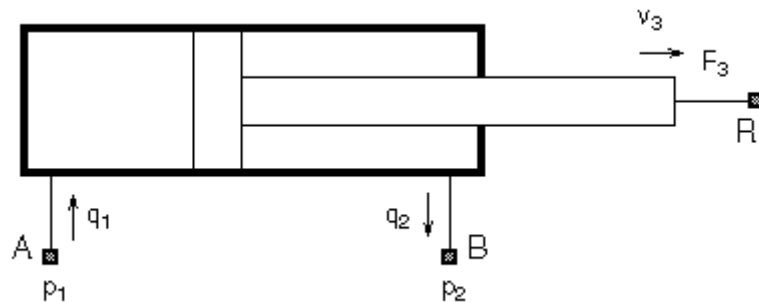
These connection ports are bidirectional. They mimic physical connections between elements. Connecting Simscape blocks together is analogous to connecting real components, such as pumps, valves, and so on. In other words, Simscape diagrams mimic the physical system layout. If physical components can be connected, their models can be connected, too. You do not have to specify flow directions and information flow when connecting Simscape blocks, just as you do not have to specify this information when you connect

real physical components. The Physical Network approach, with its Through and Across variables and bidirectional physical connections, automatically resolves all the traditional issues with variables, directionality, and so on.

The number of connection ports for each element is determined by the number of energy flows it exchanges with other elements in the system, and depends on the level of idealization. For example, a fixed-displacement hydraulic pump in its simplest form can be represented as a two-port element, with one energy flow associated with the inlet (suction) and the other with the outlet. In this representation, the angular velocity of the driving shaft is assumed constant, making it possible to neglect the energy exchange between the pump and the shaft. To account for a variable driving torque, you need a third port associated with the driving shaft.

An energy flow is characterized by its variables. Each energy flow is associated with two variables, one Through and one Across (see “Variable Types” on page 1-7 for more information). Usually, these are the variables whose product is the energy flow in watts. They are called the basic, or conjugate, variables. For example, the basic variables for mechanical translational systems are force and velocity, for mechanical rotational systems—torque and angular velocity, for hydraulic systems—flow rate and pressure, for electrical systems—current and voltage.

The following example illustrates a Physical Network representation of a double-acting hydraulic cylinder.



The element is represented with three energy flows: two flows of hydraulic energy through the inlet and outlet of the cylinder and a flow of mechanical

energy associated with the rod motion. It therefore has the following three connector ports:

- A — Hydraulic conserving port associated with pressure p_1 (an Across variable) and flow rate q_1 (a Through variable)
- B — Hydraulic conserving port associated with pressure p_2 (an Across variable) and flow rate q_2 (a Through variable)
- R — Mechanical translational conserving port associated with rod velocity v_3 (an Across variable) and force F_3 (a Through variable)

See “Connector Ports and Connection Lines” on page 1-11 for more information on connector port types.

Variable Types

Simscape supports two types of variables:

- Through — Variables that are measured with a gauge connected in series to an element.
- Across — Variables that are measured with a gauge connected in parallel to an element.

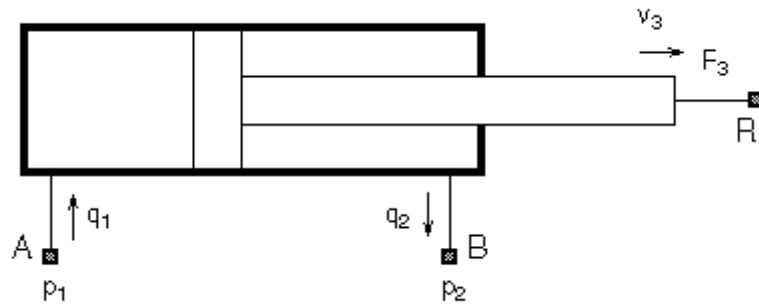
The following table contains examples of Through and Across variables specific to hydraulic, mechanical, electrical, and thermal systems:

Technology	Across Variables	Through Variables
Mechanical	Position Linear velocity Linear acceleration Angle Angular velocity Angular acceleration	Force Torque
Hydraulic	Pressure	Flow rate Volume

Technology	Across Variables	Through Variables
Electrical	Voltage Flux	Current Charge
Thermal	Temperature	Heat flow Enthalpy Entropy

Building the Mathematical Model

Through and Across variables associated with all the energy flows form the basis of the mathematical model of the block.



For example, the model of a double-acting hydraulic cylinder shown in the previous illustration can be described with a simple set of equations:

$$F_3 = p_1 A_1 - p_2 A_2$$

$$q_1 = A_1 v_3$$

$$q_2 = A_2 v_3$$

where

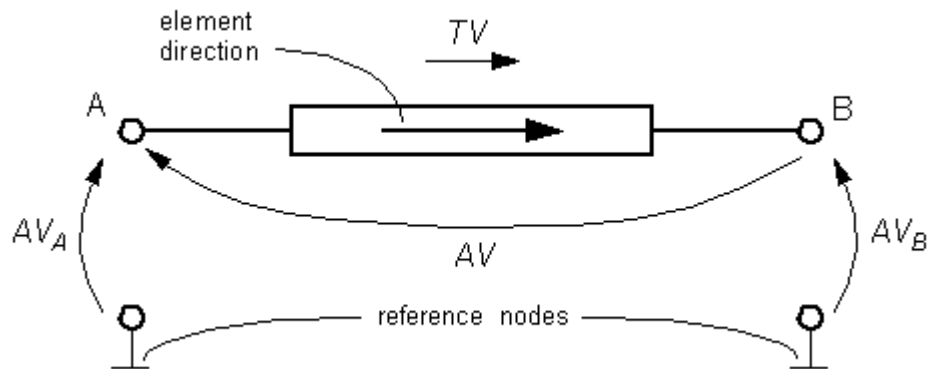
q_1, q_2	Flow rates through ports A and B, respectively (Through variables)
p_1, p_2	Gauge pressures at ports A and B, respectively (Across variables)
A_1, A_2	Piston effective areas
F_3	Rod force (Through variable)
v_3	Rod velocity (Across variable)

The model could be considerably more complex, for example, it could account for friction, fluid compressibility, inertia of the moving parts, and so on. For all these different mathematical models, however, the element configuration (that is, the number and type of ports and the associated Through and Across variables) would remain the same, meaning that the Physical Network approach lets you substitute models of different levels of complexity without introducing any changes to the schematic. For example, you can start developing your system by using the Resistive Tube block from the Foundation library, which accounts only for friction losses. At a later stage in development, you may want to account for fluid compressibility. You can then replace it with a Hydraulic Pipeline block, available with SimHydraulics, or, depending on your application, even with a Segmented Pipeline block if you also need to account for fluid inertia. This modeling principle is called incremental modeling.

Direction of Variables

Each variable is characterized by its magnitude and sign. The sign is the result of measurement orientation. The same variable can be positive or negative, depending on the polarity of a measurement gauge. That is why it is very important to apply exactly the same rule to all the variables in the Physical Network.

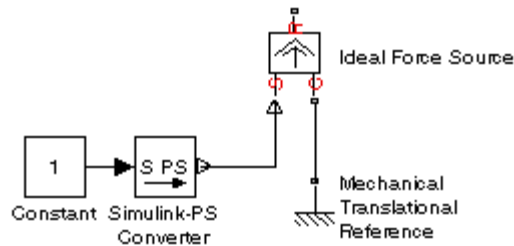
Elements with only two ports are characterized with one pair of variables, a Through variable and an Across variable. Since these variables are closely related, their orientation is defined with one direction. For example, if an element is oriented from port A to port B, it implies that the Through variable (TV) is positive if it “flows” from A to B, and the Across variable is determined as $AV = AV_A - AV_B$, where AV_A and AV_B are the element node potentials or, in other words, the values of this Across variable at ports A and B, respectively.



This approach to the direction of variables has the following benefits:

- Provides a simple and consistent way to determine whether an element is active or passive. Energy is one of the most important characteristics to be determined during simulation. If the variables direction, or sign, is determined as described above, their product (that is, the energy) is positive if the element consumes energy, and is negative if it provides energy to a system. This rule is followed throughout Simscape.
- Simplifies the model description. Symbol $A \rightarrow B$ is enough to specify variable polarity for both the Across and the Through variables.
- Lets you apply the oriented graph theory to network analysis and design.

As an example of variables direction rules, let us consider the Ideal Force Source block. In this block, as in many other mechanical blocks, port C is associated with the source reference point (case), and port R is associated with the rod.



The block positive direction is from port C to port R. This means that the force is positive if it acts in the direction from C to R, and causes bodies connected to port R to accelerate in the positive direction. The relative velocity is determined as $v = v_C - v_R$, where v_R , v_C are the absolute velocities at ports R and C, respectively, and it is negative if velocity at port R is greater than that at port C. The power generated by the source is computed as the product of force and velocity, and is negative if the source provides energy to the system.

All the elements in a network are divided into active and passive elements, depending on whether they deliver energy to the system or dissipate (or store) it. Active elements (force and velocity sources, flow rate and pressure sources, etc.) must be oriented strictly in accordance with the line of action or function that they are expected to perform in the system, while passive elements (dampers, resistors, springs, pipelines, etc.) can be oriented either way.


Connector Ports and Connection Lines

Simscape blocks may have the following types of ports:

- Physical Conserving ports — Bidirectional hydraulic or mechanical ports that represent physical connections and relate physical variables based on the Physical Network approach.
- Physical Signal ports — Unidirectional ports transferring signals that use an internal Simscape engine for computations.

Each of these ports and connections between them are described in greater detail below.

Physical Conserving Ports

Simscape blocks have special Conserving ports . You connect Conserving ports with Physical connection lines, distinct from normal Simulink lines. Physical connection lines have no inherent directionality and represent the exchange of energy flows, according to the Physical Network approach.


- You can connect Conserving ports only to other Conserving ports of the same type.
- The Physical connection lines that connect Conserving ports together are bidirectional lines that carry physical variables (Across and Through variables, as described above) rather than signals. You cannot connect Physical lines to Simulink ports or to Physical Signal ports.
- Two directly connected Conserving ports must have the same values for all their Across variables (such as pressure or angular velocity).
- You can branch Physical connection lines. When you do so, components directly connected with one another continue to share the same Across variables. Any Through variable (such as flow rate or torque) transferred along the Physical connection line is divided among the multiple components connected by the branches. How the Through variable is divided is determined by the system dynamics.

For each Through variable, the sum of all its values flowing into a branch point equals the sum of all its values flowing out.

The following table lists the types of Physical Conserving ports used in Simscape, along with the Through and Across variables associated with each type:

Port Type	Across Variable	Through Variable
Electrical	Voltage	Current
Hydraulic	Pressure	Flow rate
Mechanical translational	Translational velocity	Force
Mechanical rotational	Angular velocity	Torque

Physical Signal Ports

Physical Signal ports  carry signals between Simscape blocks. You connect them with regular connection lines, similar to Simulink signal connections. Physical Signal ports are used in Simscape block diagrams instead of Simulink input and output ports to increase computation speed and avoid issues with algebraic loops. Unlike Simulink signals, which are essentially unitless, physical signals can have units associated with them. You specify the units along with the parameter values in the block dialogs, and Simscape performs the necessary unit conversion operations when solving a physical network.

Simscape contains a Physical Signals block library. These blocks perform math operations and other functions on physical signals, and allow you to graphically implement equations inside the Physical Network.

Connecting Simscape Diagrams to Simulink Sources and Scopes

Simscape block diagrams use physical signals instead of regular Simulink signals. Therefore, you need converter blocks to connect Simscape diagrams to Simulink sources and scopes.

Use the Simulink-PS Converter block to connect Simulink sources or other Simulink blocks to the inputs of a Physical Network diagram. You can also use it to specify the input signal units. For more information, see the Simulink-PS Converter block reference page.

Use the PS-Simulink Converter block to connect outputs of a Physical Network diagram to Simulink scopes or other Simulink blocks. You can also use it to specify the desired output signal units. For more information, see the PS-Simulink Converter block reference page.

For an example of using converter blocks to connect Simscape diagrams to Simulink sources and scopes, see “Creating a Simple Model” on page 2-7.

Working with Physical Units

Unlike Simulink signals, which are essentially unitless, physical signals can have units associated with them. You specify the units along with the parameter values in the block dialogs, and Simscape performs the necessary unit conversion operations when solving a physical network. Simscape blocks support standard measurement systems. The default block units are meter-kilogram-second or MKS (SI).

This section covers the following topics:

- “Unit Definitions” on page 1-14
- “Specifying Units in Block Dialogs” on page 1-17

Unit Definitions

Simscape unit names are defined in the `pm_units.m` file, which is shipped with the product. You can open this file to see how the Simscape physical units are defined, and also as an example when adding your own units. This file is located in the directory `matlabroot\toolbox\physmod\pm_util\pm_util`.

Default registered units and their abbreviations are listed in the following table. Use the `pm_getunits` command to get an up-to-date list of units currently defined in your unit registry. Use the `pm_adddimension` and `pm_addunit` commands to define additional units.

Physical Unit Abbreviations Defined in Simscape

Quantity	Abbreviation	Unit
Angle	rad	Radian
	deg	Degree
	rev	Revolution
Angular velocity	rpm	Revolutions/minute
Capacitance	F	Farad
Charge	C	Coulomb
Current	A	Ampere

Physical Unit Abbreviations Defined in Simscape (Continued)

Quantity	Abbreviation	Unit
Energy	J	Joule
	Btu	British Thermal Unit
Flow rate	lpm	Liter/minute
	gpm	Gallon/minute
Force	N	Newton
	dyn	Dyne
	lbf	Pound-force
Inductance	H	Henry
Length	m	Meter
	cm	Centimeter
	mm	Millimeter
	km	Kilometer
	in	Inch
	ft	Foot
	mi	Mile
	yd	Yard
Mass	kg	Kilogram
	g	Gram
	mg	Milligram
	lbm	Pound mass
	oz	Ounce
	slug	Slug

Physical Unit Abbreviations Defined in Simscape (Continued)

Quantity	Abbreviation	Unit
Pressure	Pa	Pascal
	bar	Bar
	atm	Atmosphere
	psi	Pound/inch^2
Power	W	Watt
	HP	Horsepower
Resistance	Ohm	Ohm
Temperature	K	Kelvin
Time	s	Second
	min	Minute
	hr	Hour
Velocity	mph	Miles/hour
	fpm	Feet/minute
	fps	Feet/second
Viscosity absolute	Poise	Poise
	cP	Centipoise
	reyn	Reyn
Viscosity kinematic	St	Stokes
	cSt	Centistokes
	Newt	Newt

Physical Unit Abbreviations Defined in Simscape (Continued)

Quantity	Abbreviation	Unit
Volume	l	Liter
	gal	Gallon
Voltage	V	Volt

Note This table lists the unit abbreviations defined in the product. See the following section, “Specifying Units in Block Dialogs” on page 1-17, for information on how to use the abbreviations above, or mathematical expressions with these abbreviations, to specify units for the parameter values in the block dialogs.

Specifying Units in Block Dialogs

Most of the block dialogs have pull-down menus of units next to a parameter value. For example, in the Constant Volume Chamber block, the pull-down menu for the **Chamber volume** parameter contains l, gal, in³, ft³, mm³, cm³, m³, and km³, and the pull-down menu for the **Initial pressure** parameter contains Pa, bar, psi, and atm.

The converter blocks (Simulink-PS Converter and PS-Simulink Converter) let you type a unit name, or a mathematical expression with unit names, in the **Unit** field of the block dialog. You can use the abbreviations for the units defined in your registry, or any valid mathematical expressions with these abbreviations. For example, you can specify torque in newton-meters (N*m) or pound-feet (lbf*ft). To specify velocity, you can use one of the defined unit abbreviations (mph, fpm, fps), or an expression based on any combination of the defined units of length and time, such as meters/second (m/s), millimeters/second (mm/s), inches/minute (in/min), and so on.

The following operators are supported in the unit mathematical expressions:

- * Multiplication
- / Division

- ^ Power
- +, - Plus, minus — for exponents only
- () Brackets to specify evaluation order

Metric unit prefixes, such as *kilo*, *milli*, or *micro*, are not supported. For example, if you want to use milliliter as a unit of volume, you have to add it to the unit registry:

```
pm_addunit('ml', 0.001, 'l');
```

Note Currently, physical units are not propagated through the blocks in the Physical Signals library, such as PS Add, PS Gain, and so on.

Modeling Physical Systems

This chapter introduces you to modeling physical systems with Simscape. After reviewing the essential rules of connecting blocks, it moves you through a series of short tutorials. The later examples build cumulatively on the earlier ones but are presented in a way that allows you to learn from one without studying the others.

Introducing the Simscape Block Libraries (p. 2-2)

Overview of the library structure

Essential Steps to Building a Physical Model (p. 2-5)

Describes general modeling techniques

Creating a Simple Model (p. 2-7)

Step-by-step example of creating a simple Simscape model

Modeling Best Practices (p. 2-25)

How to use best practices in your model architecture to avoid numerical simulation issues

Introducing the Simscape Block Libraries

This section shows you how to open and use the block libraries in Simscape:

- “Library Structure Overview” on page 2-2
- “Using the Simulink Library Browser to Access the Block Libraries” on page 2-2
- “Using the Command Prompt to Access the Block Libraries” on page 2-3

Library Structure Overview

Simscape contains two main libraries:

- Foundation library — Contains basic hydraulic, mechanical, electrical, and physical signal blocks
- Utilities library — Contains essential environment blocks for creating Physical Networks models

In addition, if you have installed any of the vertical products of the Physical Modeling family, such as SimHydraulics, SimDriveline, or SimMechanics, you will see the corresponding libraries under Simscape.

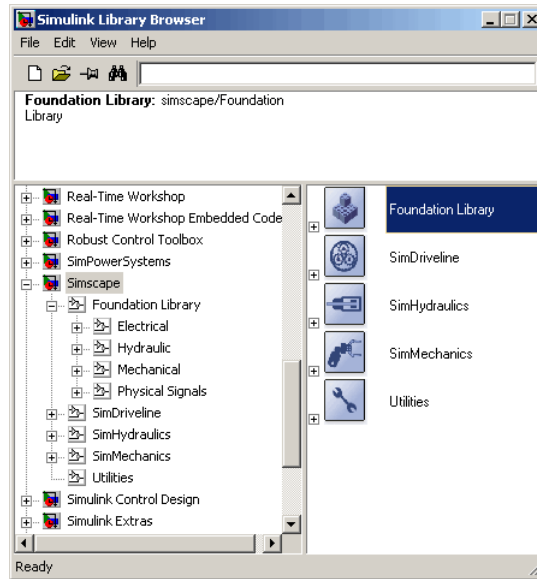
You can combine all these blocks in your Simscape diagrams to model physical systems. You can also use the basic Simulink blocks in your diagrams, such as sources or scopes. See “Connecting Simscape Diagrams to Simulink Sources and Scopes” on page 1-13 for more information on how to do this.

Using the Simulink Library Browser to Access the Block Libraries

On Microsoft Windows, you can access the blocks through the Simulink Library Browser. To display the Library Browser, click the **Library Browser** button in the toolbar of the MATLAB desktop or Simulink model window:



Alternatively, you can type `simulink` in the MATLAB Command Window. Then expand the Simscape entry in the contents tree.



Using the Command Prompt to Access the Block Libraries

On any platform:



- To open the Simscape library, type `simscape` in the MATLAB Command Window.
- To open the main Simulink library (to access generic Simulink blocks), type `simulink` in the MATLAB Command Window.

The Simscape library consists of two top-level libraries, Foundation and Utilities. In addition, if you have installed any of the vertical products of the Physical Modeling family, such as SimHydraulics, SimDriveline, or SimMechanics, you will see the corresponding libraries under Simscape, as shown in the following illustration. Some of these libraries contain second-level and third-level sublibraries. You can expand each library by double-clicking its icon. For more details on library hierarchy and descriptions of block categories, see “Blocks — By Category”.



Essential Steps to Building a Physical Model

The rules that you must follow when building a physical model with Simscape are described in “Basic Principles of Modeling Physical Networks” on page 1-5. This section briefly reviews these rules.

- Simscape blocks, in general, feature both Conserving ports  and Physical Signal inports and outports .
- There are four types of Physical Conserving ports used in Simscape: hydraulic, electrical, mechanical translational, and mechanical rotational. Each type has specific Through and Across variables associated with it.
- You can connect Conserving ports only to other Conserving ports of the same type.
- The Physical connection lines that connect Conserving ports together are bidirectional lines that carry physical variables (Across and Through variables, as described above) rather than signals. You cannot connect Physical lines to Simulink ports or to Physical Signal ports.
- Two directly connected Conserving ports must have the same values for all their Across variables (such as voltage or angular velocity).
- You can branch Physical connection lines. When you do so, components directly connected with one another continue to share the same Across variables. Any Through variable (such as current or torque) transferred along the Physical connection line is divided among the multiple components connected by the branches. How the Through variable is divided is determined by the system dynamics.

For each Through variable, the sum of all its values flowing into a branch point equals the sum of all its values flowing out.

- You can connect Physical Signal ports to other Physical Signal ports with regular connection lines, similar to Simulink signal connections. These connection lines carry physical signals between Simscape blocks.
- You can connect Physical Signal ports to Simulink ports through special converter blocks. Use the Simulink-PS Converter block to connect Simulink outports to Physical Signal inports. Use the PS-Simulink Converter block to connect Physical Signal outports to Simulink inports.

- Unlike Simulink signals, which are essentially unitless, Physical Signals can have units associated with them. SimHydraulics block dialogs let you specify the units along with the parameter values, where appropriate. Use the converter blocks to associate units with an input signal and to specify the desired output signal units.

For examples of applying these rules when creating an actual physical model, see “Creating a Simple Model” on page 2-7.

The MathWorks recommends that you build, simulate, and test your model incrementally. Start with an idealized, simplified model of your system, simulate it, verify that it works the way you expected. Then incrementally make your model more realistic, factoring in effects such as friction loss, motor shaft compliance, hard stops, and the other things that describe real-world phenomena. Simulate and test your model at every incremental step. Use subsystems to capture the model hierarchy, and simulate and test your subsystems separately before testing the whole model configuration. This approach helps you keep your models well organized and makes it easier to troubleshoot them.

Creating a Simple Model

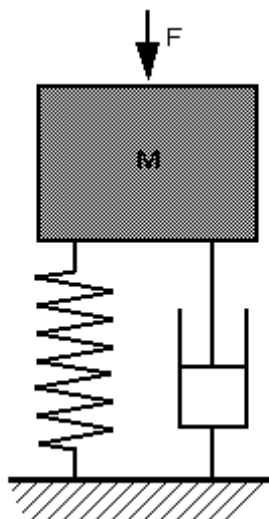
In this example, you are going to model a simple mechanical system and observe its behavior under various conditions. This tutorial illustrates the essential steps to building a physical model, described in the previous section, and makes you familiar with using the basic Simscape blocks.

The tutorial consists of the following tasks:

- “Building a Simscape Diagram” on page 2-7
- “Modifying Initial Settings” on page 2-15
- “Running the Simulation” on page 2-16
- “Adjusting the Parameters” on page 2-19

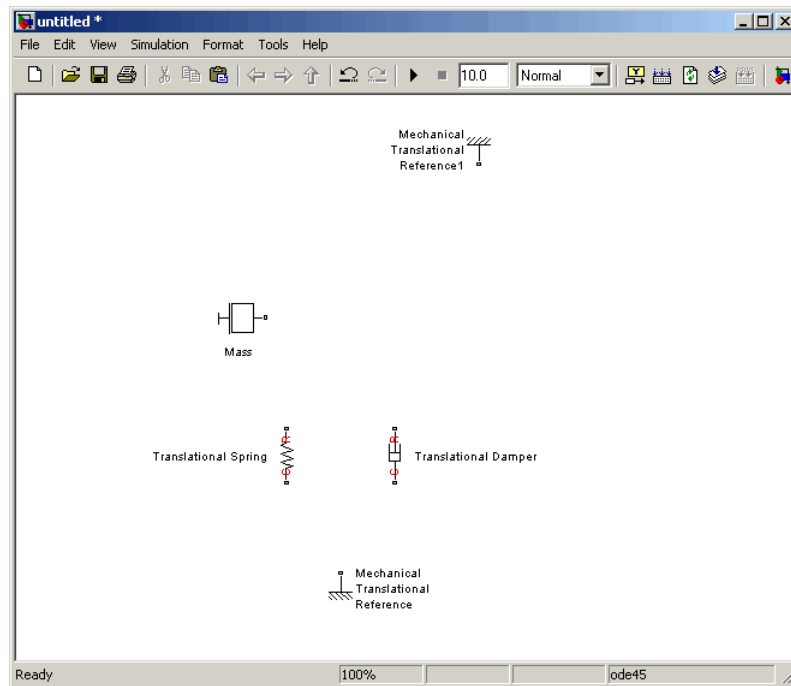
Building a Simscape Diagram

The following schematic represents a simple model of a car suspension. It consists of a spring and damper connected to a body (represented as a mass), which is agitated by a force. You can vary the model parameters, such as the stiffness of the spring, the mass of the body, or the force profile, and view the resulting changes to the velocity and position of the body.

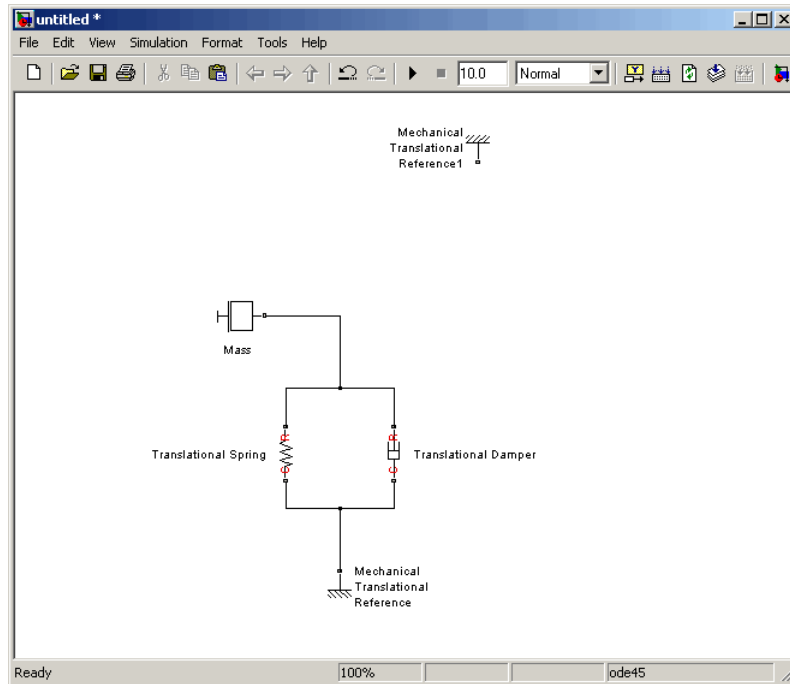


To create an equivalent Simscape diagram, follow these steps:

- 1 Open the Simscape and Simulink block libraries, as described in “Introducing the Simscape Block Libraries” on page 2-2.
- 2 Create a new Simulink model. To do this, click the **New** button on the Library Browser’s toolbar (Windows only) or choose **New** from the library window’s **File** menu and select **Model**. Simulink creates an empty model in memory and displays it in a new model editor window.
- 3 Open the Simscape > Foundation Library > Mechanical > Translational Elements library.
- 4 Drag the Mass, Translational Spring, Translational Damper, and two Mechanical Translational Reference blocks into the model window.
- 5 Orient the blocks as shown in the following illustration. To rotate a block, select it and press **Ctrl+R**.

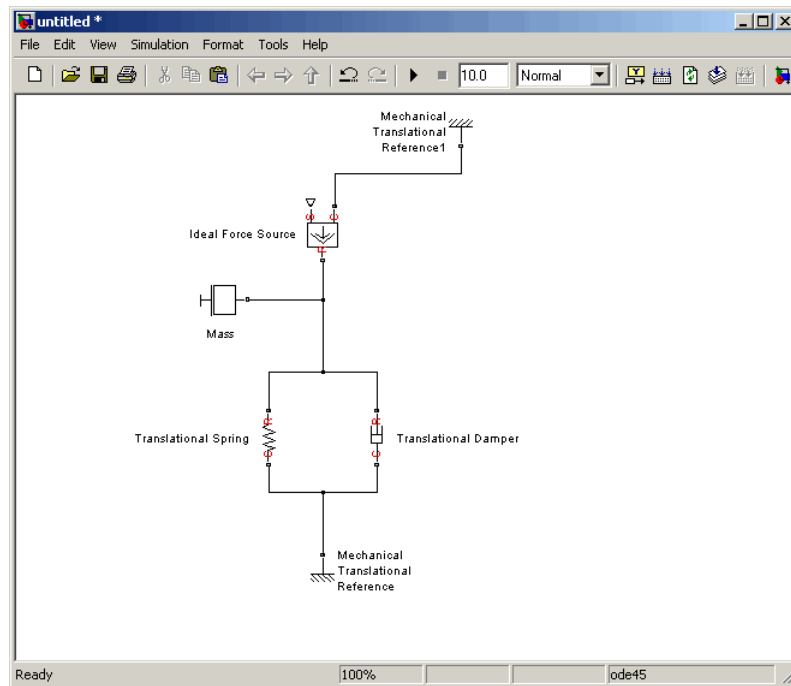


- 6 Connect the Translational Spring, Translational Damper, and Mass blocks to one of the Mechanical Translational Reference blocks as shown in the next illustration.

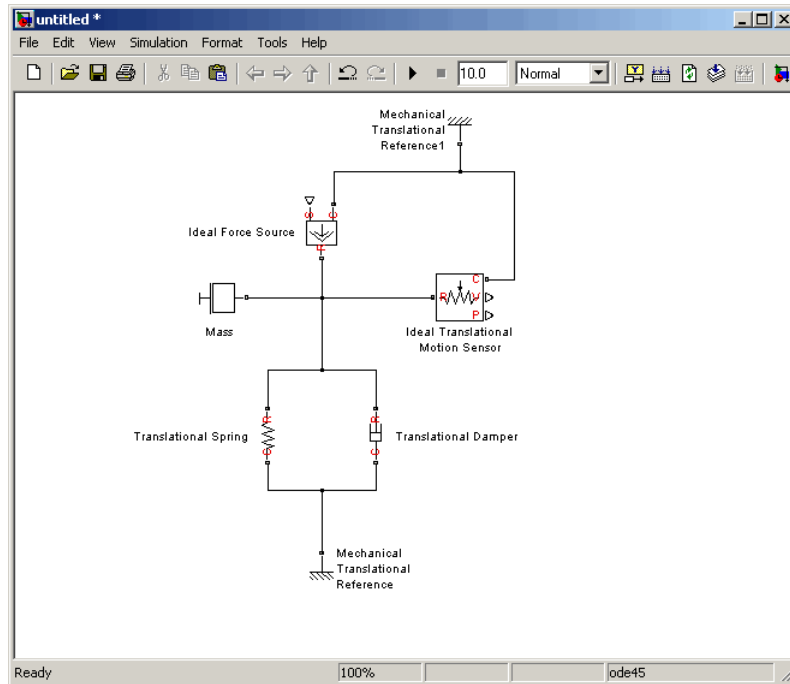


- 7 To add the representation of the force acting on the mass, open the Simscape > Foundation Library > Mechanical > Mechanical Sensors and Sources library and add the Ideal Force Source block to your diagram.

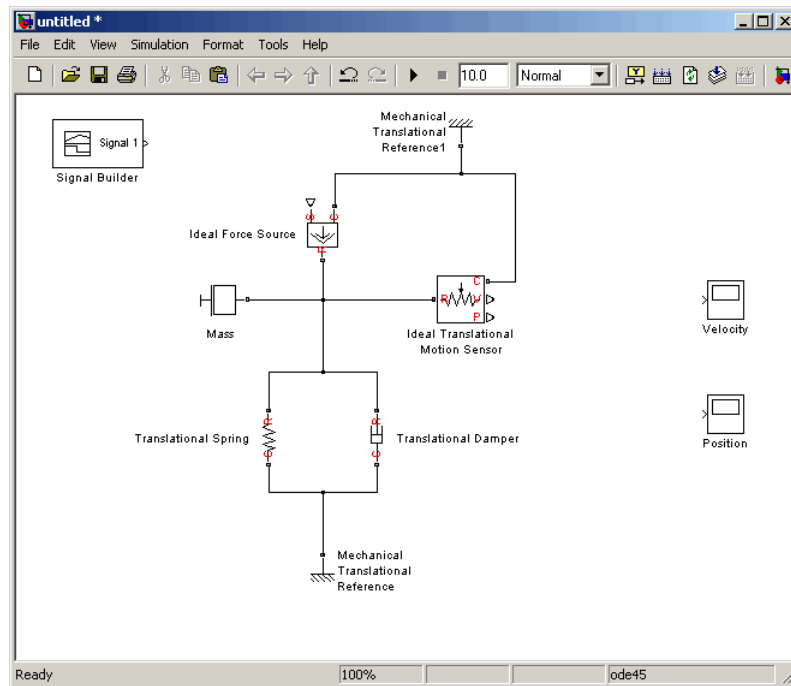
To reflect the correct direction of the force shown in the original schematic, flip the block by selecting **Format > Flip Block** from the top menu bar of the model window. Connect the block's port C (for “case”) to the second Mechanical Translational Reference block, and its port R (for “rod”) to the Mass block, as shown below.



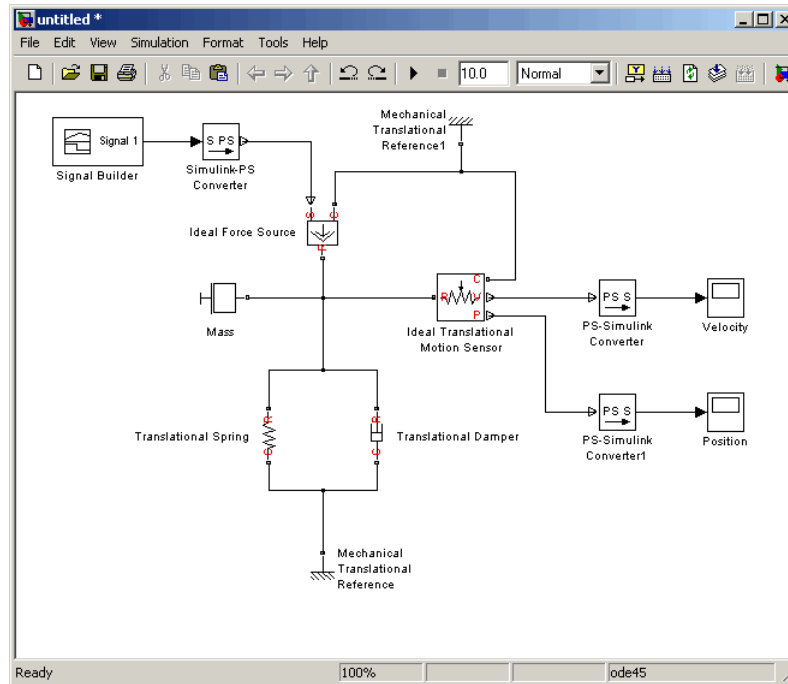
- 8** Add the sensor to measure speed and position of the mass. Place the Ideal Translational Motion Sensor block from the Mechanical Sensors and Sources library into your diagram and connect it as shown below.



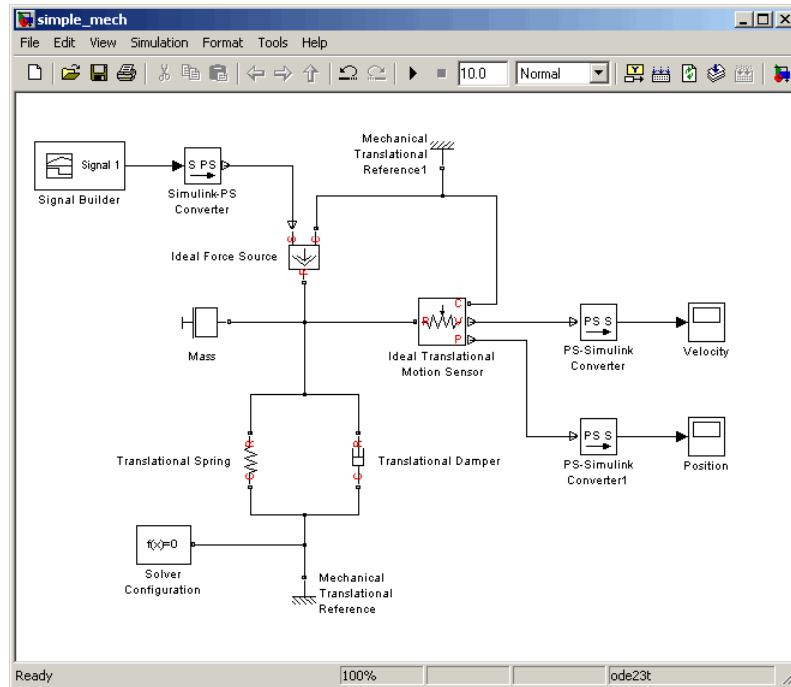
- 9 Now you need to add the sources and scopes. They are found in the regular Simulink libraries. Open the Simulink > Sources library and copy the Signal Builder block into the model. Then open the Simulink > Sinks library and copy two Scope blocks. Rename one of the Scope blocks to Velocity and the other to Position.



- 10** Every time you connect a Simulink source or scope to a Simscape diagram, you have to use an appropriate converter block, to convert Simulink signals into physical signals and vice versa. Open the Simscape > Utilities library and copy a Simulink-PS Converter block and two PS-Simulink Converter blocks into the model. Connect the blocks as shown below.



- 11 Each topologically distinct physical network in a diagram requires exactly one Solver Configuration block, found in the Simscape > Utilities library. Copy this block into your model and connect it to the circuit by creating a branching point and connecting it to the only port of the Solver Configuration block. Your diagram now should look like this.



12 Your block diagram is now complete. Save it as `simple_mech.mdl`.

Modifying Initial Settings

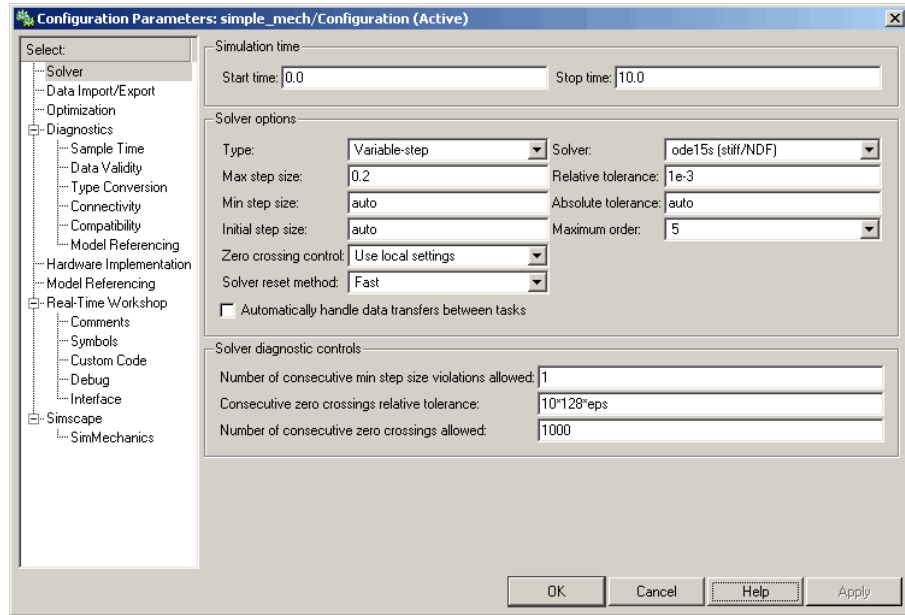
After you have put together a block diagram of your model, as described in the previous section, you need to select a solver and provide the correct values for configuration parameters.

To prepare for simulating the model, follow these steps:

- 1** Select a Simulink solver. On the top menu bar of the model window, select **Simulation > Configuration Parameters**. The Configuration Parameters dialog box opens, showing the **Solver** node.

Under **Solver options**, set **Solver** to `ode15s` (Stiff/NDF) and **Max step size** to 0.2.

Also note that **Simulation time** is specified to be between 0 and 10 seconds. You can adjust this setting later, if needed.



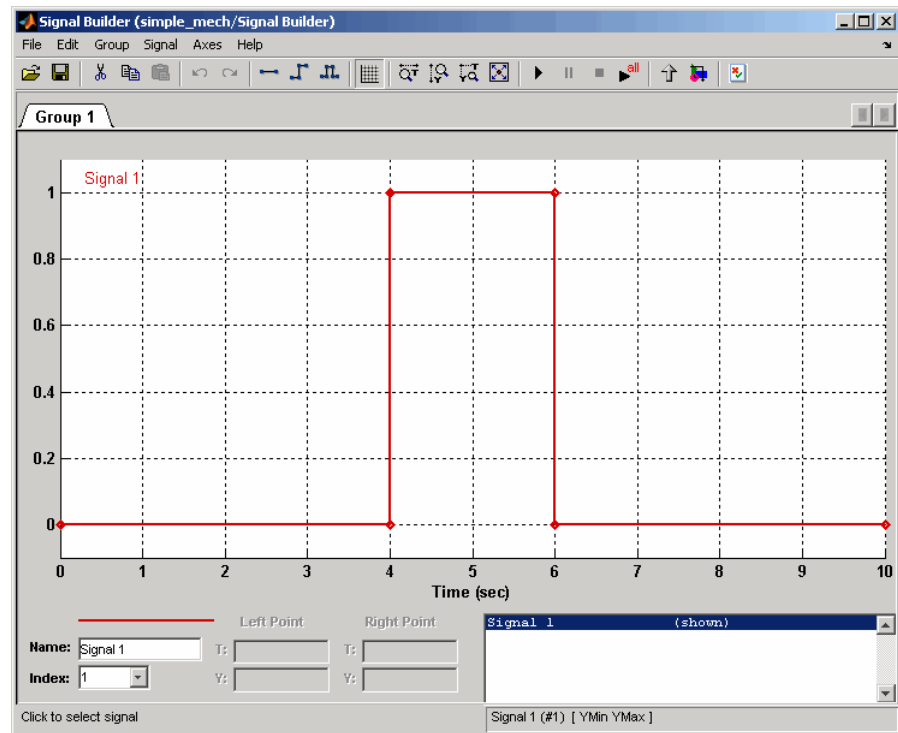
Click **OK** to close the Configuration Parameters dialog box.

2 Save the model.


Running the Simulation

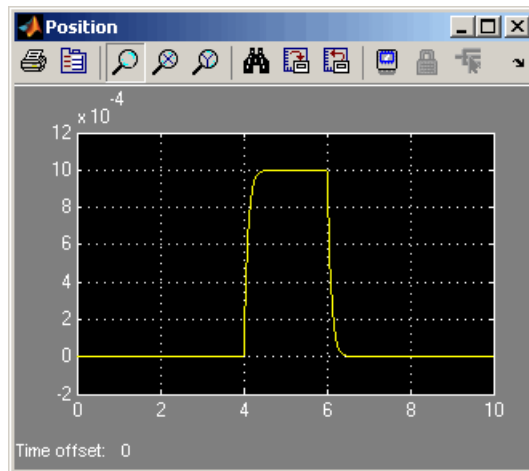
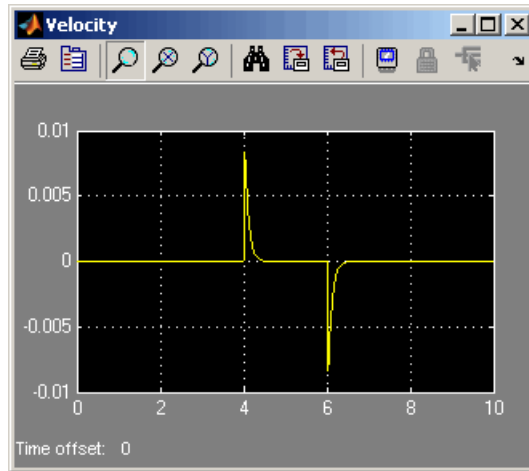
After you've put together a block diagram and specified the initial settings for your model, you can run the simulation.

- 1** The input signal for the force is provided by the Signal Builder block. The signal profile is shown in the illustration below. It starts with a value of 0, then at 4 seconds there is a step change to 1, and then it changes back to 0 at 6 seconds. This is the default profile.



The Velocity scope outputs the mass velocity, and the Position scope outputs the mass displacement as a function of time. Double-click both scopes to open them.

- 2 To run the simulation, click  in the model window toolbar. The Simscape solver evaluates the model, calculates the initial conditions, and runs the simulation. For a detailed description of this process, see “How Simscape Works” on page 3-2. Completion of this step may take a few seconds. The message in the bottom-left corner of the model window provides the status update.
- 3 Once the simulation starts running, the Velocity and Position scope windows display the simulation results, as shown in the next illustration.



In the beginning, the mass is at rest. Then at 4 seconds, as the input signal changes abruptly, the mass velocity spikes in the positive direction and gradually returns to zero. The mass position at the same time changes more gradually, on account of inertia and damping, and stays at the new value as long as the force is acting upon it. At 6 seconds, when the input signal changes back to zero, the velocity gets a mirror spike, and the mass gradually returns to its initial position.

You can now adjust various inputs and block parameters and see their effect on the mass velocity and displacement.

Adjusting the Parameters

After running the initial simulation, you can experiment with adjusting various inputs and block parameters.

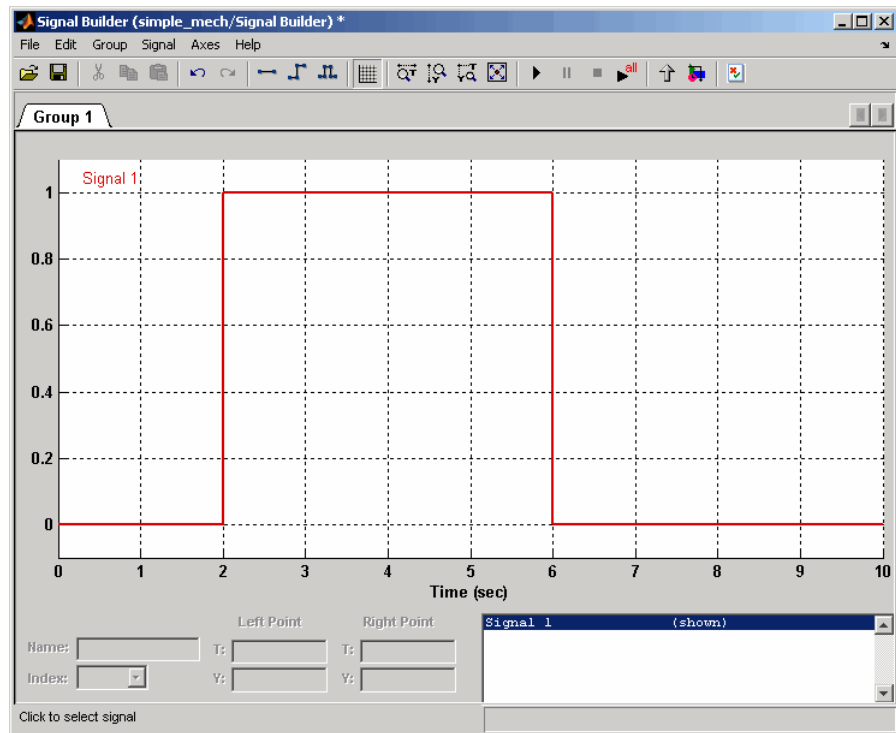
Try the following adjustments:

- 1** Change the force profile.
- 2** Change the model parameters.
- 3** Change the mass position output units.

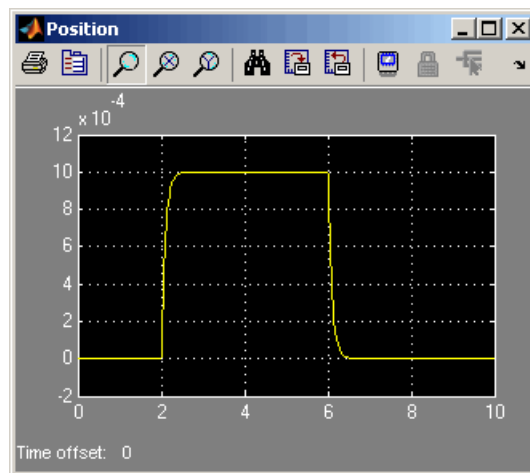
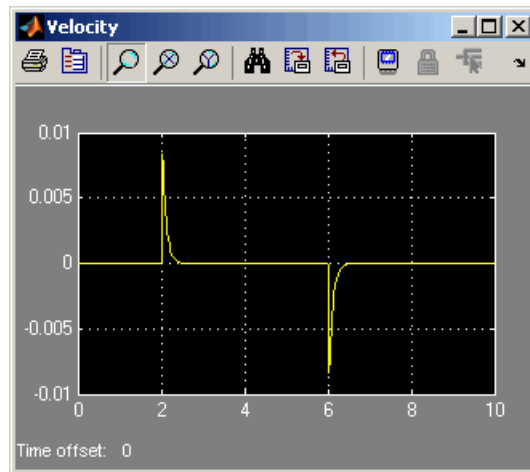
Changing the Force Profile

This example shows how a change in the input signal affects the force profile, and therefore the mass displacement.

- 1** Double-click the Signal Builder block to open it.
- 2** Click the first vertical segment of the signal profile and drag it from 4 to 2 seconds, as shown below. Close the block dialog.



- 3** Run the simulation. The simulation results are shown in the following illustration.

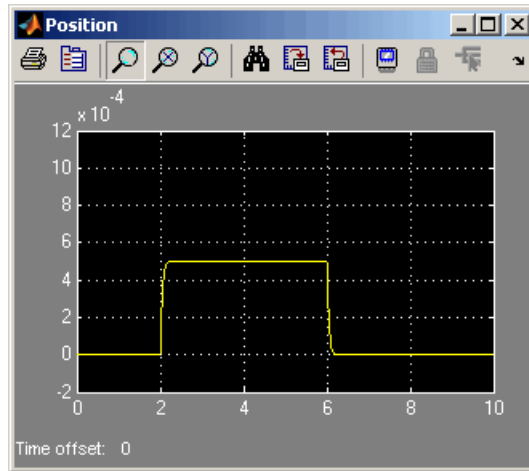


Changing the Model Parameters

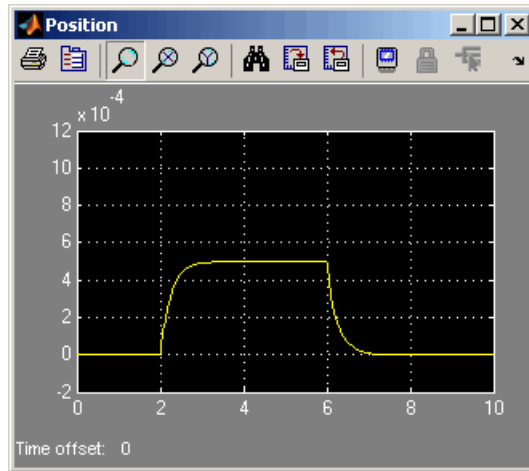
In our model, the force acts on a mass against a translational spring and damper, connected in parallel. This example shows how changes in the spring stiffness and damper viscosity affect the mass displacement.

- 1 Double-click the Translational Spring block. Set its **Spring rate** to 2000 N/m.

- 2 Run the simulation. The increase in spring stiffness results in smaller amplitude of mass displacement, as shown in the following illustration.




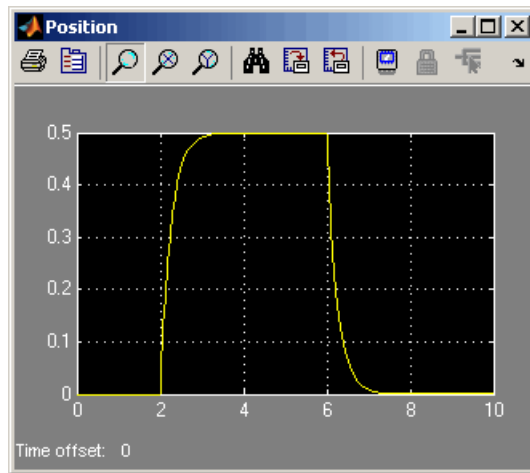
- 3 Next, double-click the Translational Damper block. Set its **Damping coefficient** to 500 N/(m/s).
- 4 Run the simulation. Because of the increase in viscosity, the mass is slower both in reaching its maximum displacement and in returning to the initial position, as shown in the following illustration.



Changing the Mass Position Output Units

In our model, we have used the PS-Simulink Converter block in its default parameter configuration, which does not specify units. Therefore, the Position scope outputs the mass displacement in the default length units, that is, in meters. This example shows how to change the output units for the mass displacement to millimeters.

- 1 Double-click the PS-Simulink Converter block. Type mm in the **Unit** text box and click **OK**.
- 2 Run the simulation. In the Position scope window, click  to autoscale the scope axes. The mass displacement is now output in millimeters, as shown in the following illustration.



Modeling Best Practices

Certain configurations of physical modeling blocks can cause numerical difficulties or slow down your simulation. When this happens, Simscape issues a warning in the MATLAB workspace and, if it fails to initialize, a Simscape error.

In electrical circuits, common examples that can cause this behavior include capacitors in parallel, inductors in series, voltage sources in parallel and current sources in series. Often, the cause of the numerical difficulty is immediately apparent. For example, two capacitors in parallel must have identical voltage values, otherwise the ports connecting them would not be physical conserving ports. In practical circuits, topologies such as parallel capacitors are possible, and small difference in their instantaneous voltages is possible due to parasitic series resistance.

Note Mathematically, these topologies result in the so-called *Index-2 Differential Algebraic Equations* (DAEs). Their solution requires two differentiations of the constraint equations, and as such it is numerically better to avoid these component topologies where possible.

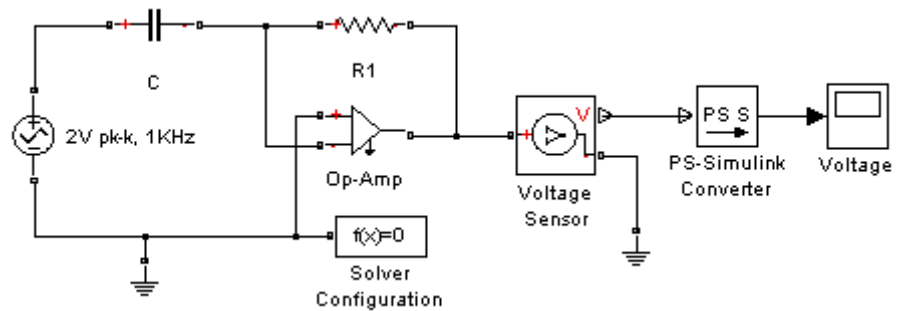
There are two approaches to resolving these difficulties. The first is to change the circuit to an equivalent simpler one. For example, two parallel capacitors can be replaced by a single equivalent capacitor, and likewise for two series inductors. For the capacitors, the individual currents for the two original capacitors can be calculated from the total current and the two capacitance values, if required. Similarly, the voltages across two individual series inductors can be inferred from voltage across the single equivalent plus knowledge of the two inductance values. For some circuit topologies, it is not possible to find an equivalent simpler one that resolves the problem, and the second approach is needed.

The second approach is to include small parasitic resistances in the component. In the Simscape Foundation library, the Capacitor and Inductor blocks include such parasitic terms so that you can connect capacitances in parallel and inductors in series. If your circuit does not have any such topologies, then you can change the default parasitic terms to zero. Note that other blocks do not contain these parasitic terms, for example, the

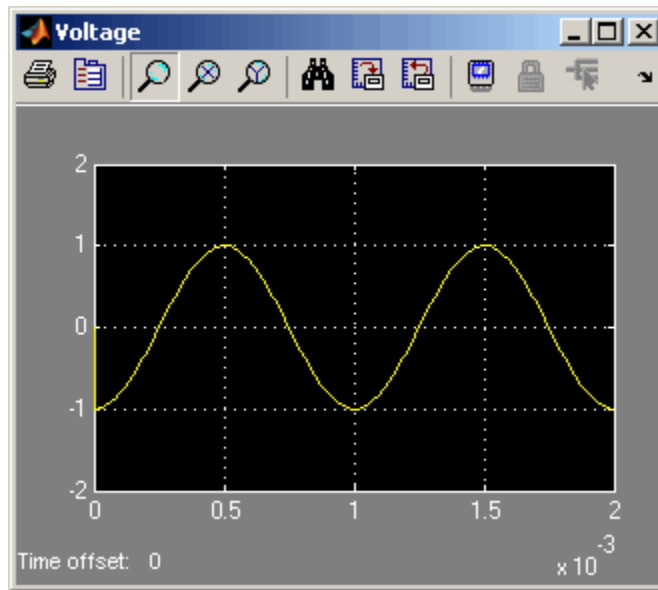
Mutual Inductor block. Therefore, if you wanted to connect two mutual inductor primaries in series, you would need to introduce your own parasitic conductance across at least one of the primary windings.

Example of Using a Parasitic Resistance to Avoid Numerical Simulation Issues

The following diagram models a differentiator that might be used as part of a Proportional-Integral-Derivative (PID) controller. You can open this model by typing `ssc_differentiator` in the MATLAB Command Window.



Simulate the model, and you will see that the output is minus the derivative of the input sinusoid.



Now open the capacitor C block dialog, and set the series resistance to zero. The model now runs very slowly, and issues a warning:

```
Warning: problems possible for transient initialization, as well as stepsize control
for transient solve, due to equations of one or more components:
'ssc_differentiator/2V pk-k, 1KHz'
'ssc_differentiator/Op-Amp'
'ssc_differentiator/C'
```

```
Problem diagnosed in check before start of transient simulation.
```

The cause of the warning is that the circuit effectively connects the voltage source in parallel with the Op-Amp input. This is because an ideal op-amp satisfies $V_+ = V_-$, where V_+ and V_- are the noninverting and inverting inputs, respectively. This is an example where it is not possible to replace the circuit with an equivalent simpler one, and a parasitic small resistance has to be introduced.

Simulating Physical Models

Simscape gives you multiple ways to simulate and analyze physical systems in the Simulink environment. Running a physical model simulation is similar to running a simulation of any other Simulink model. It entails setting various simulation options, starting the simulation, and viewing the simulation results. See the Using Simulink documentation for a general discussion of these topics. This chapter focuses on aspects of simulation specific to Simscape and SimHydraulics models. Refer to the SimMechanics and SimDriveline documentation for specifics of simulating and analyzing SimMechanics and SimDriveline models.

How Simscape Works (p. 3-2)	Explains how Simscape validates and simulates a model
Linearizing Simscape Models (p. 3-7)	Describes specifics of using Simulink linearization commands in Simscape
Limitations (p. 3-9)	Describes restrictions and limitations on using Simulink tools in Simscape

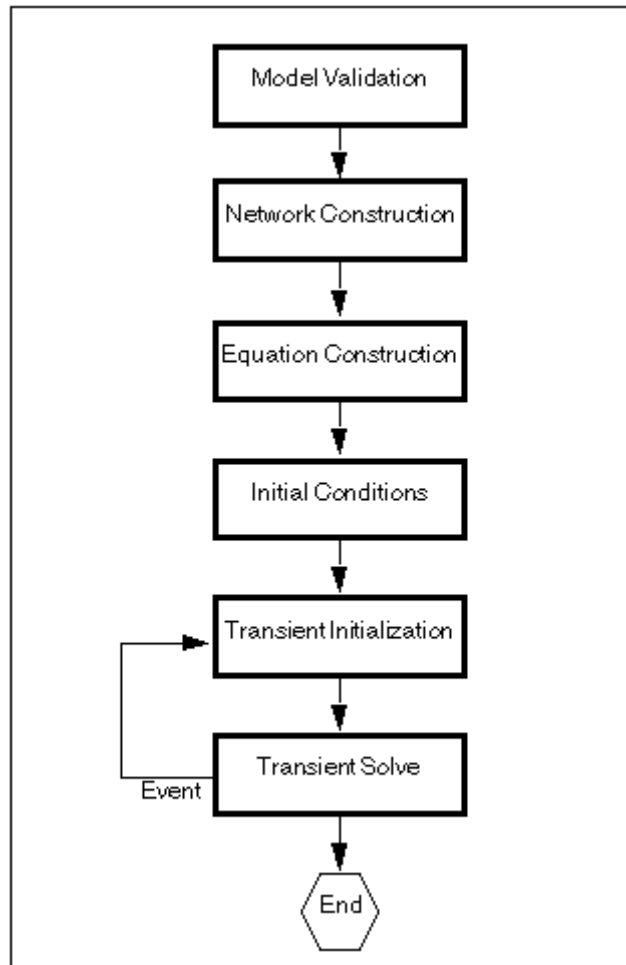
How Simscape Works

You might find this brief overview of how Simscape works helpful for constructing models and understanding errors.

Simscape simulation consists of the following major phases:

- 1** “Model Validation” on page 3-3
- 2** “Network Construction” on page 3-4
- 3** “Equation Construction” on page 3-4
- 4** “Computing Initial Conditions” on page 3-5
- 5** “Performing Transient Initialization” on page 3-6
- 6** “Transient Solve” on page 3-6

Simscape simulation sequence is shown in the following flow chart.



Model Validation

Simscape first validates the model configuration and checks your data entries from the block dialogs. In particular:

- Each topologically distinct physical network in a diagram requires exactly one Solver Configuration block.

- If your model contains hydraulic elements, each topologically distinct hydraulic circuit in a diagram requires exactly one Custom Hydraulic Fluid block (or Hydraulic Fluid block, available with SimHydraulics) to be connected to it. Therefore, there must be as many hydraulic fluid blocks as there are hydraulic circuits in the system. These blocks define the fluid properties that act as global parameters for all the blocks connected to the hydraulic circuit.
- Signal units specified in a Simulink-PS Converter block must match the input type expected by the Simscape block connected to it. For example, when you provide the input signal for an Ideal Angular Velocity Source block, specify angular velocity units, such as rad/s or rpm, in the Simulink-PS Converter block, or leave it unitless. Similarly, units specified in a PS-Simulink Converter block must match the type of physical signal provided by the Simscape block output.

Network Construction

After validating the model, Simscape constructs the physical network based on the following principles:

- Two directly connected Conserving ports have the same values for all their Across variables (such as voltage or angular velocity).
- Any Through variable (such as current or torque) transferred along the Physical connection line is divided among the multiple components connected by the branches. For each Through variable, the sum of all its values flowing into a branch point equals the sum of all its values flowing out.

Equation Construction

Based on the network configuration, the parameter values provided in the block dialogs, and the global parameters defined by the fluid properties, if applicable, Simscape constructs the system of equations for the model.

These equations contain variables of the following types:

- *Dynamic* — Time derivative of this variable appears in equations. Dynamic variables are the independent states for simulation.

- *Algebraic* — Time derivative of this variable does not appear in equations. Algebraic variables are always dependent (on dynamic variables, other algebraic variables, or inputs).

Computing Initial Conditions

Simscape computes the initial conditions only once, in the beginning of simulation ($t=0$).

Initial conditions are computed by setting all dynamic variables to 0, except those corresponding to blocks that have an initial condition field in their block dialogs, and solving for all the system variables. The blocks with initial conditions have their dynamic variables set according to the user-provided value in the block dialog. Initial conditions can only be set on dynamic variables, because these are the independent states for simulation. For example, the Translational Spring block has the **Initial deformation** parameter, so the corresponding spring position state is set to the initial offset specified in the block dialog. Refer to the block reference documentation to find which blocks have initial conditions specified through their dialogs.

This concludes the initial conditions computation when the **Start simulation from steady state** check box in the Solver block dialog box is not selected (this is the default).

When this box is selected, the solver attempts to find the steady state that would result if the inputs to the system were held constant for a sufficiently large time, starting from the initial state obtained from the initial conditions computation, described previously. Although the solver tries to find the particular steady state resulting from the given initial conditions, it is not guaranteed to do so. All that is guaranteed is that if the steady-state solve succeeds, the state found is a steady state (within tolerance). Steady state means that the system variables are no longer changing with time. Simulation then starts from this steady state.

Note If the simulation fails at or near the start time when you use the **Start simulation from steady state** option, consider clearing the check box and simulating with the plain initial conditions computation only.

Performing Transient Initialization

After computing the initial conditions, or after a subsequent event (such as a zero-crossing or discontinuity), Simscape performs transient initialization. This is done by fixing all dynamic variables and solving for algebraic variables and derivatives of dynamic variables. The goal of transient initialization is to provide a consistent set of initial conditions for the next transient solve step. At this phase, the initial conditions at the beginning of simulation are no longer enforced, but the dynamic state of the system is preserved (or, in other words, the energy in the system is preserved).

Transient Solve

Finally, Simscape performs transient solve of the system of equations. In transient solve, continuous differential equations are integrated in time to compute all the variables as a function of time.

Simscape continues to perform the simulation according to the results of the transient solve until it encounters an event, such as a zero crossing or discontinuity. The event may be within the physical network or elsewhere in the Simulink model. If an event is encountered, Simscape returns to the phase of transient initialization, and then back to transient solve. This cycle continues until the end of simulation.

Linearizing Simscape Models

The Simulink `linmod` and `dlinmod` commands create continuous- or discrete-time linear time-invariant (LTI) state-space models from Simulink models. You can use these commands to generate an LTI state-space model from a model containing Simscape components.

There are two basic ways `linmod` and `dlinmod` can be used, and the behavior of linearization differs depending on which method is chosen. If `linmod(md1)` is called (that is, the arguments for time, state, and input are not provided), then consistent initial conditions are solved for in the same way as on the first step of a simulation. In particular, any initial conditions, such as initial offset from equilibrium for a spring, are set just as if the simulation were starting from the initial time.

However, if `linmod(md1, t, x, u)` is called, it does not solve for the start-of-simulation initial conditions. An important difference of Simscape from Simulink is that the states in Simscape are not generally independent (some states have algebraic dependencies on other states). Thus, it is possible, though erroneous, to provide `linmod` with an inconsistent state to linearize about. If the state is not consistent (within a suitably chosen tolerance based on the solver tolerance), then Simscape issues a warning at the command line when linearization is attempted. Simscape will then attempt to make the provided `x` consistent by changing some of its components, possibly by large amounts. Thus it is important, when providing a state to `linmod` and `dlinmod`, to ensure that the state provided is reasonably consistent. This is most easily done by taking the state from some time in a simulation run, for example, by using the **States** check box on the **Data Import/Export** pane of the Configuration Parameters dialog box, accessed by selecting **Simulation > Configuration Parameters** from the top menu bar.

Note If the **Enable automatic function scaling** check box in the Solver block dialog box is selected (or cleared) when you obtain a state through simulation to provide it to `linmod` or `dlinmod`, this check box must remain selected (or cleared) when `linmod` or `dlinmod` is called.

The technique used to obtain linear models involves obtaining an independent subset of the system states as the states for linearization. These independent

state variables are a subset of the dynamic variables of the system. Thus the A matrix, of size n_states by n_states , is all zeros except for a submatrix of size n_ind by n_ind , where n_ind is the number of independent states. The B matrix, of size n_states by n_inputs , is all zeros except for a submatrix of size n_ind by n_inputs . The C matrix, of size $n_outputs$ by n_states , is all zeros except for a submatrix of size $n_outputs$ by n_ind . The D matrix, of size $n_outputs$ by n_inputs , may have nonzeros everywhere. To obtain a model where the number of states equals the number of independent states, use the function `smnreal` in Control System Toolbox.

Limitations

Certain Simulink tools and features are not supported in Simscape.

- “Solver Support” on page 3-9
- “Signal Support in the Simulink-PS Converter Block” on page 3-9
- “Algebraic Loops” on page 3-9
- “Code Generation” on page 3-10

Solver Support

Explicit solvers are not currently supported in Simscape. Only the following solvers are supported:

- ode15s
- ode23t
- ode14x

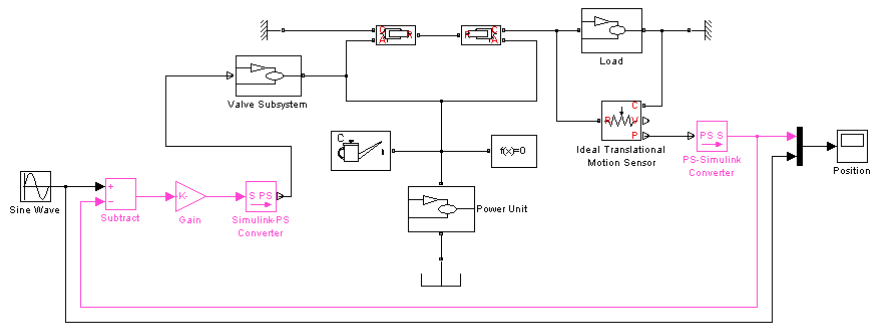
Signal Support in the Simulink-PS Converter Block

The Simulink-PS Converter block currently supports only continuous sample-time signals as its input. If you use a discrete sample-time signal, you may get incorrect results.

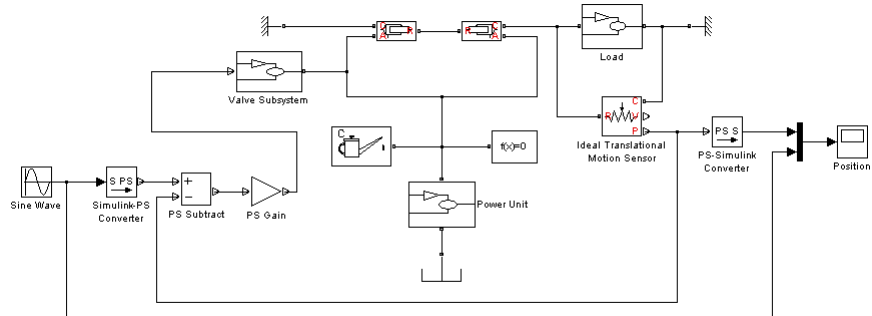
Algebraic Loops

A Simscape physical network should not exist within a Simulink algebraic loop. This means that you should not directly connect an output of a PS-Simulink Converter block to an input of a Simulink-PS Converter block of the same physical network.

For example, the following model contains a direct feedthrough between the PS-Simulink Converter block and the Simulink-PS Converter block (highlighted in magenta). To avoid the algebraic loop, you can insert a Transfer Function block anywhere along the highlighted loop.



A better way to avoid an algebraic loop without introducing additional dynamics is shown in the modified model below.



Code Generation

Code generation is not currently supported for all Simscape products. You can generate code for models that contain only SimMechanics or SimDriveline blocks. If a model contains SimHydraulics or foundation Simscape blocks, code generation is not supported.

Using the Simscape Editing Mode

About the Simscape Editing Mode
(p. 4-2)

Working with Restricted and Full
Modes (p. 4-9)

Editing Mode Information (p. 4-23)

Introduces the Simscape Editing
Mode

Describes how to perform common
design and simulation tasks while
using the SEM functionality

Describes ways of finding
information related to using
the Editing Mode functionality

About the Simscape Editing Mode

The Simscape Editing Mode (SEM) functionality is implemented for corporate customers who perform physical modeling and simulation using Simscape and its vertical products: SimDriveline, SimHydraulics, and SimMechanics. It allows you to open, simulate, and save models that contain blocks from vertical products in Restricted mode, without checking out vertical product licenses, as long as the products are installed on your machine.

This section covers the following topics:

- “Suggested Workflows” on page 4-2
- “What You Can Do in Restricted Mode” on page 4-3
- “What You Can Do in Full Mode” on page 4-3
- “Switching Between Modes” on page 4-4
- “Working with Block Libraries” on page 4-7

Suggested Workflows

The Simscape Editing Mode functionality supports widespread use of Physical Modeling products throughout an engineering organization by making it economical for one user to develop a model and provide it to many other users.

Specifically, this feature allows a user, *model developer*, to build a model that uses Simscape and one or more vertical products and share that model with other users, *model users*. When building the model in Full mode, the model developer must have a Simscape license and the vertical product licenses for all the blocks in the model. For example, if a model combines Simscape, SimHydraulics, and SimDriveline blocks, the model developer needs to check out licenses for all three products to work with it in Full mode. Once the model is built, model users need only to check out a Simscape license to simulate the model and fine-tune its parameters in Restricted mode. As long as no structural changes are made to the model, model users can work in Restricted mode and do not need to check out vertical product licenses.

Another workflow lets multiple users, who all have Simscape licenses, share a small number of vertical product licenses by working mostly in Restricted

mode, and temporarily switching models to Full mode only when they need to perform a specific design task that requires being in Full mode.

Note The MathWorks recommends that you save all the models in Full mode before upgrading to a new version of Simulink or Simscape.

If you have saved a model in Restricted mode and, upon upgrading to a new product version, open the model and it does not run, switch it to Full mode and save. You can then again switch to Restricted mode and work without problem.

What You Can Do in Restricted Mode

When your model is open in Restricted mode, you can:

- Simulate the model.
- Inspect parameters.
- Change certain block parameters. In general, you can change numerical parameter values, but cannot change the block parameterization options. See the block reference pages for specifics.
- Generate code (currently, only in SimMechanics and SimDriveline).
- Make data logging or visualization changes.
- Add or delete regular Simulink blocks (such as sources or scopes) and appropriate connections.

For other types of changes, listed in the following section, your model has to be in Full mode. Some of these disallowed changes are impossible to make in Restricted mode (for example, Restricted parameters are grayed out in block dialog boxes). Other changes, like changing the physical topology of a model, are not explicitly disallowed, but if you make these changes in Restricted mode, Simscape will issue an error message when you try to run, compile, or save such a model.

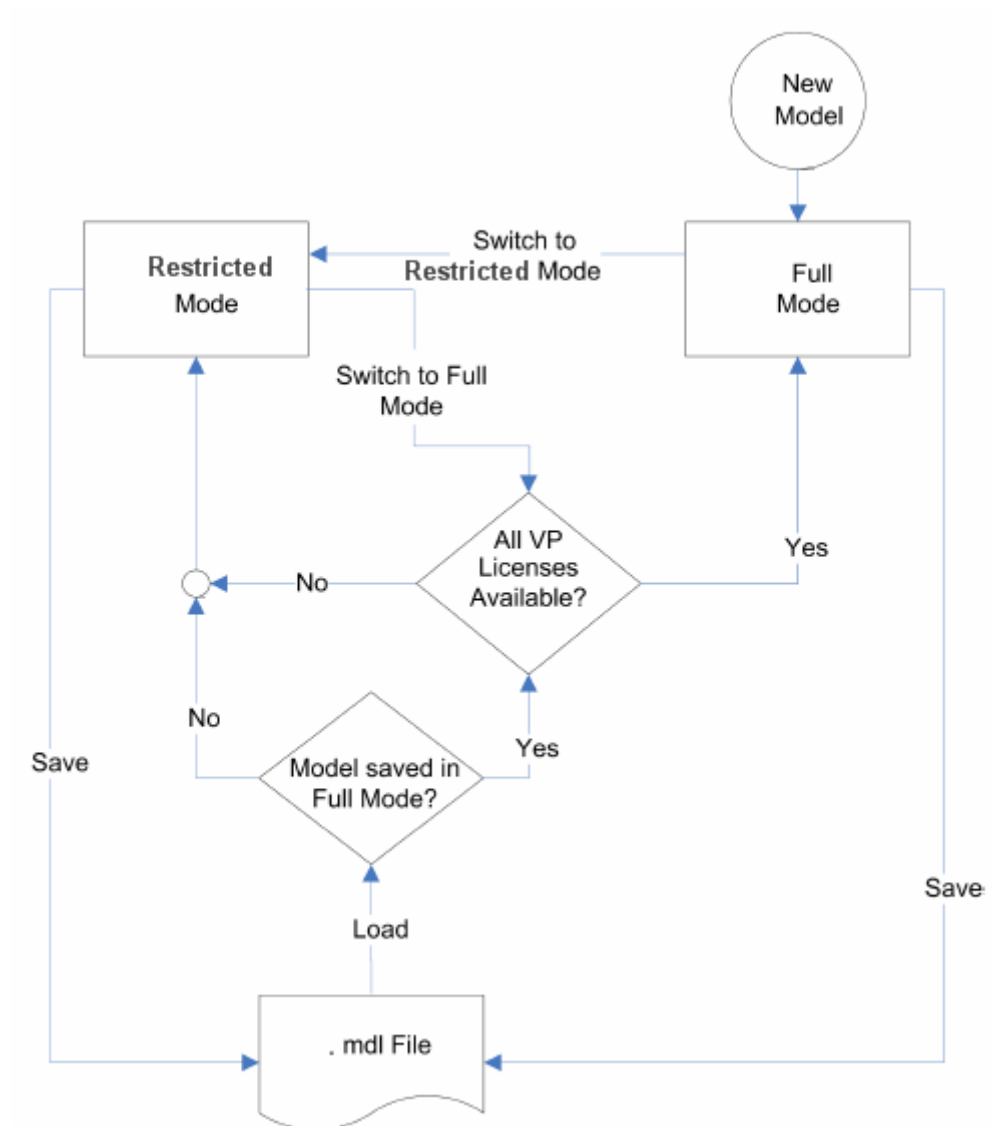
What You Can Do in Full Mode

You need to open a model in Full mode if you need to do any of the following:

- Add or delete Physical Modeling blocks (that is, Simscape blocks or blocks from the vertical product libraries).
- Make or break Physical connections (between Conserving or Physical Signal ports).
- Change the types of signals going into actuators or out of sensors (for example, from velocity to torque).
- Change configuration parameters.
- Change block parameterization options and other restricted parameters.
- Change physical units of parameters.

Switching Between Modes

The following flow chart shows what happens when you switch between modes.



New models are always created in Full mode. You can then either save the model in Full mode, or switch to Restricted mode and save the model in Restricted mode.

When you load an existing model, Simscape checks whether it has been saved in Full or Restricted mode.

- If the model has been saved in Restricted mode, it opens in Restricted mode.
- If the model has been saved in Full mode, Simscape checks whether all the vertical product licenses for this model are available and, if so, opens it in Full mode. If a vertical product license is not available, Simscape issues an error message and opens the model in Restricted mode. See also “Example with Multiple Vertical Products” on page 4-6.

Note You can set a Simulink preference to specify that the models are always to open in Restricted mode, regardless of the way they have been saved.

When a model is open, you can transition it between Full and Restricted modes at any time, in either direction:

- When you try to switch from Restricted to Full mode, Simscape checks whether all the vertical product licenses for this model are available. If a vertical product license is not available, Simscape issues an error message and the model stays in Restricted mode. See also “Example with Multiple Vertical Products” on page 4-6.
- No checks are performed when switching from Full to Restricted mode.

Note If a vertical product license has been checked out to open a model in Full mode, it remains checked out for the remainder of the MATLAB session. Switching to Restricted mode does not immediately return the license.

Example with Multiple Vertical Products

When you try to open a model in Full mode or to switch from Restricted to Full mode, Simscape scans the model and attempts to check out the required vertical product licenses as it encounters them in the model. If a license is not available, Simscape issues an error message and the model stays in Restricted mode. The licenses are checked out sequentially. As a result, if a model uses blocks from multiple vertical products, some of the vertical

product licenses may have already been checked out by the time Simscape encounters an unavailable license. In this case, these vertical product licenses stay checked out until you quit the MATLAB session, even though the model is in Restricted mode.

For example, consider a model that uses blocks from SimHydraulics and SimDriveline libraries, but the user who tries to open it has only the SimDriveline license available. It may happen that Simscape checks out a SimDriveline license first, and then tries to check out a SimHydraulics license, which is not available. Simscape then issues an error message and opens the model in Restricted mode, but the SimDriveline license stays checked out until the end of the MATLAB session.

Working with Block Libraries

This section describes the specifics of working with block libraries while using the Editing Mode functionality. These rules are applicable to any physical modeling blocks, that is, blocks from all Simscape libraries, including the vertical products. In general, you need to work in Full mode when you modify a library block. However, when you open a model that references the modified block, you may work in Restricted mode, under certain conditions. The following summary details the Editing Mode rules for modifying and using library blocks:

- To add physical modeling blocks to a library block, you need to work in Full mode.
 - If this library block had not previously contained physical modeling blocks, you need to work in Full mode to load a preexisting model that uses this library block or to drag this block to a model.
 - If this library block had previously contained physical modeling blocks, you can work in Restricted mode when loading a preexisting model that uses this library block. However, you have to work in Full mode to drag this block from the library to a model.
- To add external physical ports to a library block, you need to work in Full mode.
 - You can work in Restricted mode when loading a preexisting model that uses this library block.

- However, to connect these additional ports, you need to work in Full mode because you are changing the model topology.
- To delete external physical ports from a library block, you need to work in Full mode. If these ports were connected in a model saved in Restricted mode, loading the model causes the topology to change, so you need to switch to Full mode to save or compile the model.

Resolving Block Library Links

All Simscape blocks in your models, including the vertical products' blocks, must have resolved block library links. You can neither disable nor break these library links. This is a global requirement of Simscape, which is necessary to enforce the Editing Mode rules for modifying and using library blocks, listed above. A model with broken library links will neither compile nor save. You must restore all the broken block library links for your model to be valid.

If you want to customize certain blocks and use them in your models, you must add these modified blocks to your own custom library, then copy the block instances that you need to your model.

Working with Restricted and Full Modes

This section describes how to perform common design and simulation tasks while utilizing the Editing Mode functionality.

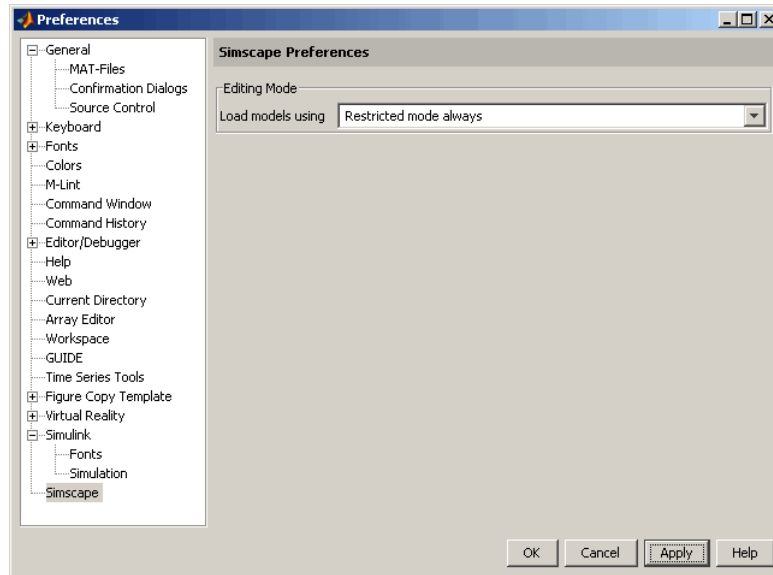
This section covers the following topics:

- “Setting the Model Loading Preference” on page 4-9
- “Saving a Model in Restricted Mode” on page 4-10
- “Working with a Model in Restricted Mode” on page 4-12
- “Switching from Restricted to Full Mode” on page 4-20

Setting the Model Loading Preference

By default, when you load an existing model, Simscape checks whether it has been saved in Full or Restricted mode and tries to open it in this mode. However, you can set your preferences so that the models are always open in Restricted mode, regardless of the way they have been saved.

- 1** From the MATLAB or Simulink top menu bar, select **File > Preferences**. The Preferences dialog box opens.
- 2** In the left pane of the Preferences dialog box, select **Simscape**. The right pane displays the **Editing Mode** group box. By default, the **Load models using** option is set to Editing mode specified in models.
- 3** Select Restricted mode always from the drop-down list, as shown, and click **OK**.

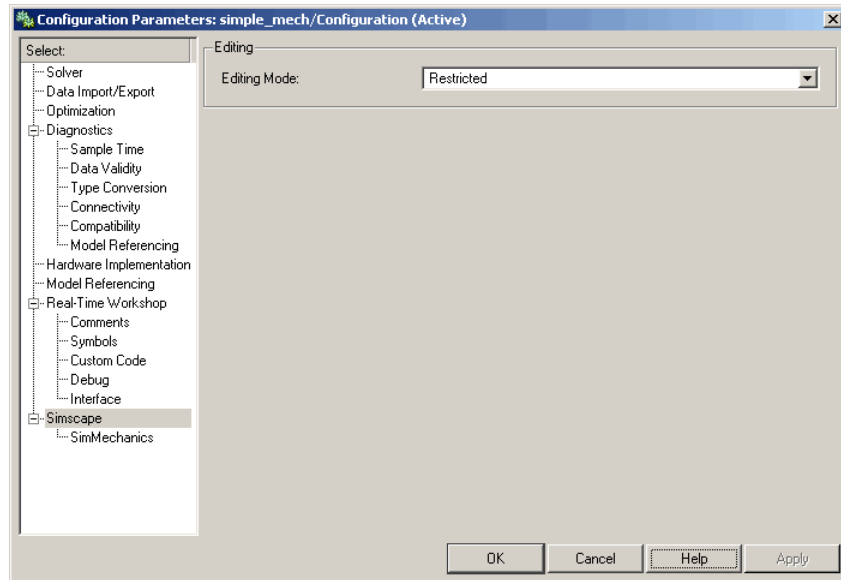


Now, when you open a model, Simscape does not attempt to check out vertical product licenses and always opens the model in Restricted mode.

Saving a Model in Restricted Mode

Rather than setting your preferences so that all the models always open in Restricted mode, you can switch an individual model to Restricted mode before saving it. Such a model will then, by default, open in Restricted mode.

- 1 From the top menu bar in the model window, select **Simulation > Configuration Parameters**. The Configuration Parameters dialog box opens.
- 2 In the left pane of the Configuration Parameters dialog box, select **Simscape**. The right pane displays the **Editing Mode** option, which is by default set to Full.
- 3 Select Restricted from the drop-down list, as shown, and click **OK**.



4 Save the model.

Note The **Simscape** entry does not appear in the left pane of the Configuration Parameters dialog box until you add at least one Physical Modeling block to your model. If you create an additional configuration set for a model, the **Simscape** entry does not appear in it until you either activate it or perform a Physical Modeling operation, such as adding or deleting a Physical Modeling block or connection, opening a Physical Modeling block dialog box, and so on.

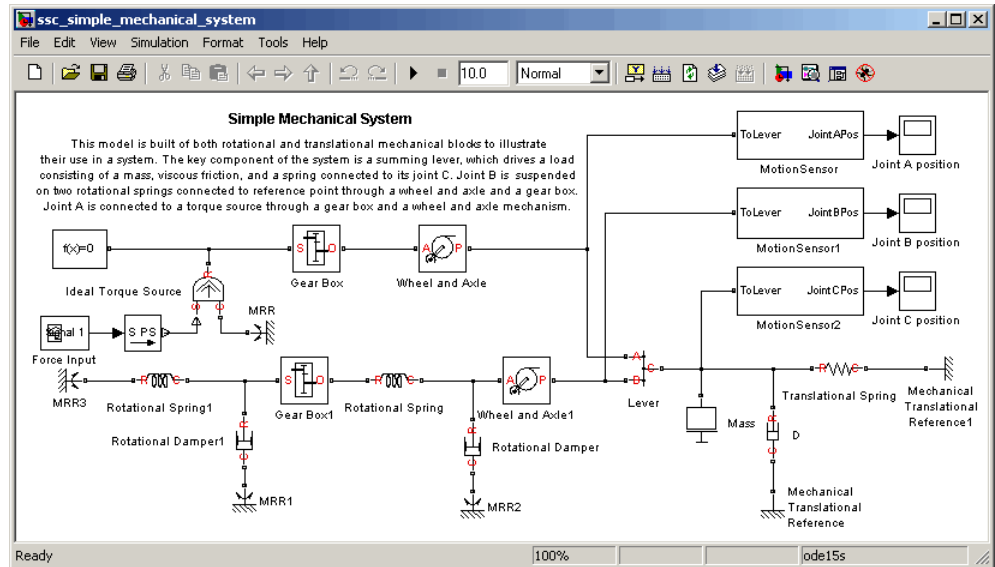
Once you have switched a model to Restricted mode, working with it follows the rules described in “Working with a Model in Restricted Mode” on page 4-12. Note, however, that the vertical product licenses for this model stay checked out until you quit the MATLAB session.

When you open a model that has been saved in Restricted mode, Simscape opens it in Restricted mode and does not check out the vertical product licenses.

Example of Saving a Model in Restricted Mode

In this example, you switch a model to Restricted mode and save it.

- 1 Open the Simple Mechanical System demo model (ssc_simple_mechanical_system).



- 2 From the top menu bar in the model window, select **Simulation > Configuration Parameters**. The Configuration Parameters dialog box opens.
- 3 In the left pane of the Configuration Parameters dialog box, select **Simscape**. The right pane displays the **Editing Mode** option, which is set to Full by default.
- 4 Select Restricted from the drop-down list and click **OK**.
- 5 Save the model as test_edit_mode.mdl.

Working with a Model in Restricted Mode

When you open a model in Restricted mode, you can perform a variety of tasks: simulate the model, inspect and fine-tune block parameters, add

and delete basic Simulink blocks, and so on. For a complete list of allowed operations, see “What You Can Do in Restricted Mode” on page 4-3.

When you open a block dialog box in Restricted mode, some of the block parameters may be grayed out. These are the so-called *restricted* parameters that can be modified only in Full mode. In general, you can change numerical parameter values in Restricted mode, but you cannot change the block parameterization options. See the block reference pages for specifics. Note also that when a restricted parameter defines the block parameterization schema, nonrestricted parameters available for fine-tuning in Restricted mode depend on the value of this restricted parameter. For example, in a Constant Volume Chamber block, the **Chamber specification** parameter is restricted. If, at the time the model entered Restricted mode, this parameter was set to *By volume*, then the nonrestricted parameters available for fine-tuning would be **Chamber volume**, **Specific heat ratio**, and **Initial pressure**. If, however, it was set to *By length and diameter*, you will have a different set of parameters available in Restricted mode.

You cannot change physical units in Restricted mode. When you open a block dialog box in Restricted mode, the drop-down lists of units next to a parameter name and value are grayed out. When you open a PS-Simulink Converter or Simulink-PS Converter block dialog box, the **Unit** parameter is grayed out.

The following examples illustrate operations allowed and disallowed in Restricted mode:

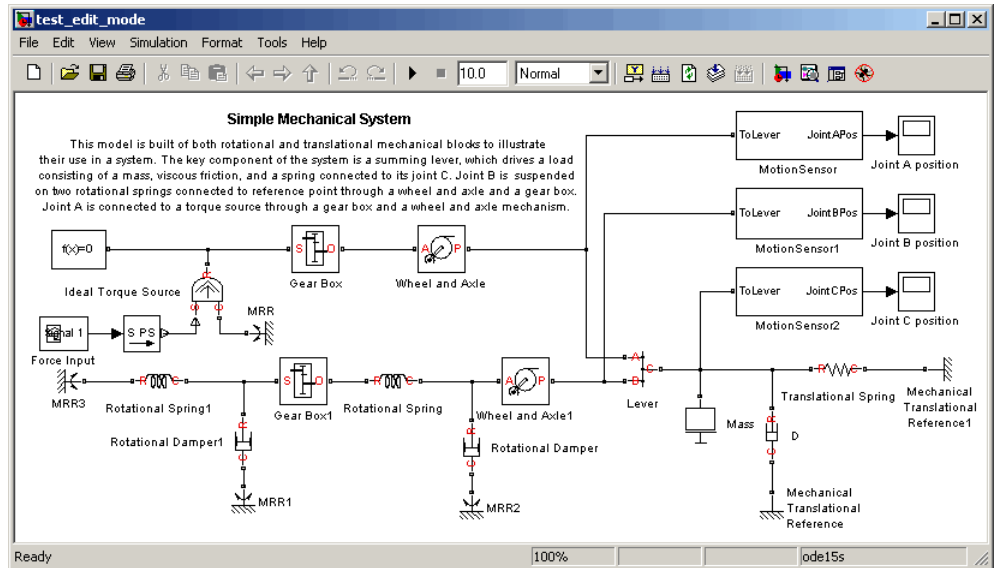
- “Simulating and Fine-Tuning a Model in Restricted Mode” on page 4-13
- “Adding and Deleting Simulink Blocks in Restricted Mode” on page 4-17
- “Performing an Operation Disallowed in Restricted Mode” on page 4-19

Simulating and Fine-Tuning a Model in Restricted Mode

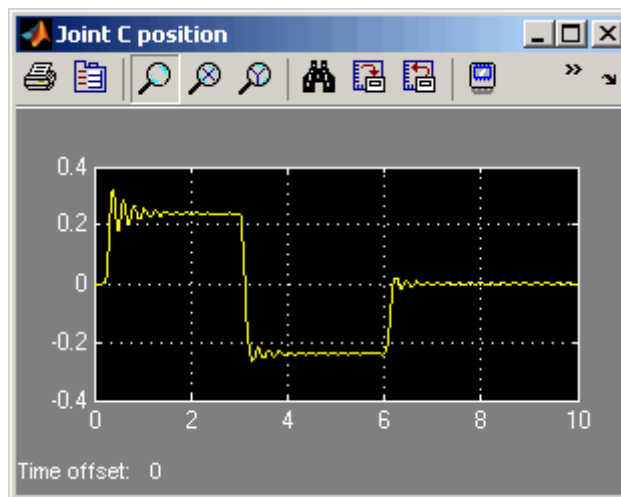
This example shows how you can work with a model in Restricted mode by changing certain parameter values and observing the simulation results.

- 1 Open the `test_edit_mode` model, which you saved in Restricted mode in “Example of Saving a Model in Restricted Mode” on page 4-12. The model opens in Restricted mode.

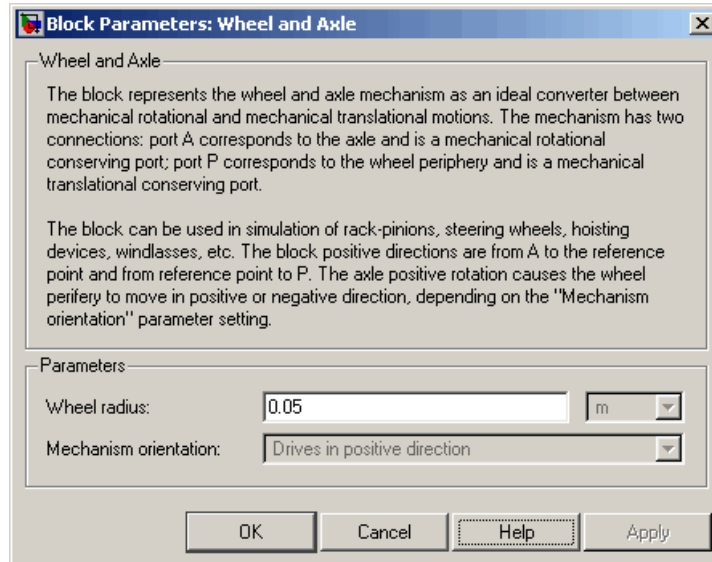
4 Using the Simscape Editing Mode



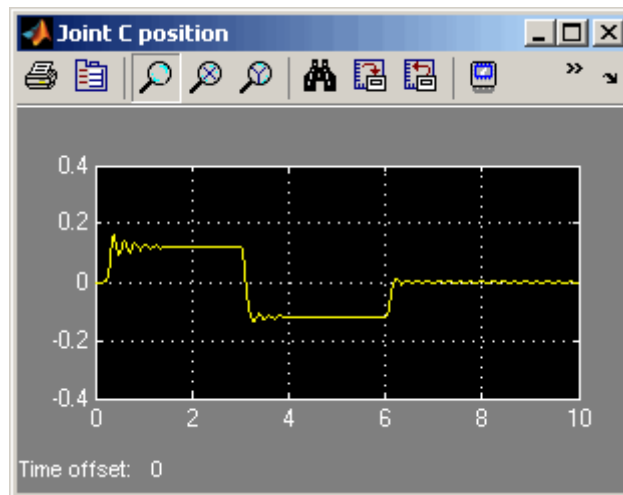
- 2 Open the Joint C Position scope and simulate the model. The models runs and simulates in Restricted mode.



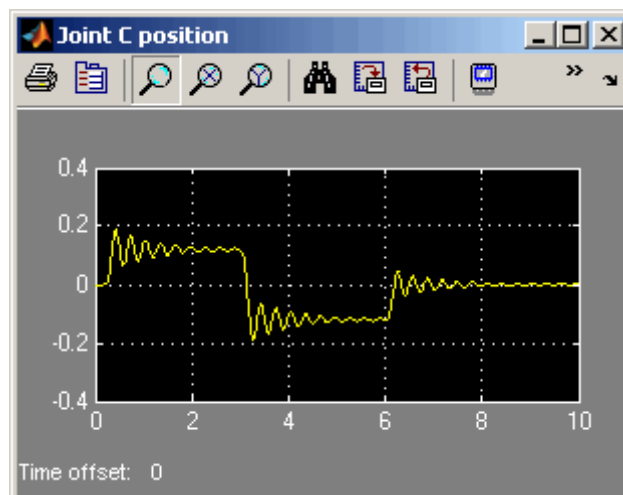
- 3 Double-click the Wheel and Axle block to open its dialog box. Notice that the **Mechanism orientation** parameter is grayed out, because you cannot modify the block driving direction in Restricted mode.



- 4 Change the **Wheel radius** parameter value to 0.1.
- 5 Simulate the model again. Notice that the motion amplitude of node C became smaller as a result of the wheel radius change.



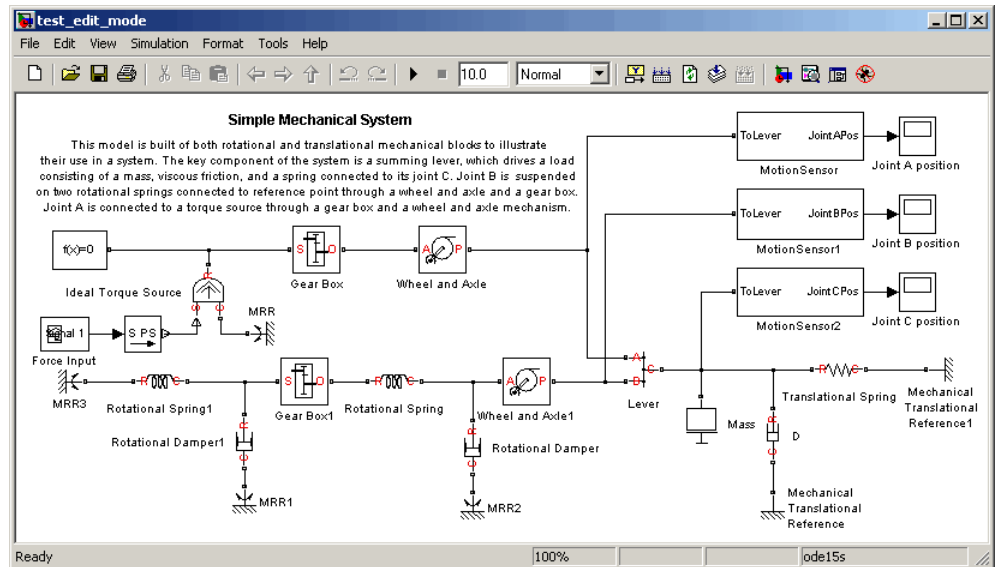
- 6 Double-click the Mass block and change the **Mass** parameter value to 24.
- 7 Simulate the model. Notice that doubling the mass resulted in increased vibrations.



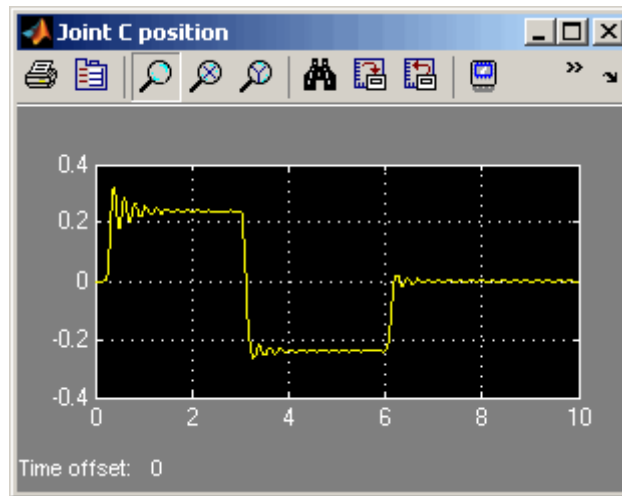
Adding and Deleting Simulink Blocks in Restricted Mode

This example shows how you can change the model input signal in Restricted mode by adding and deleting basic Simulink blocks.

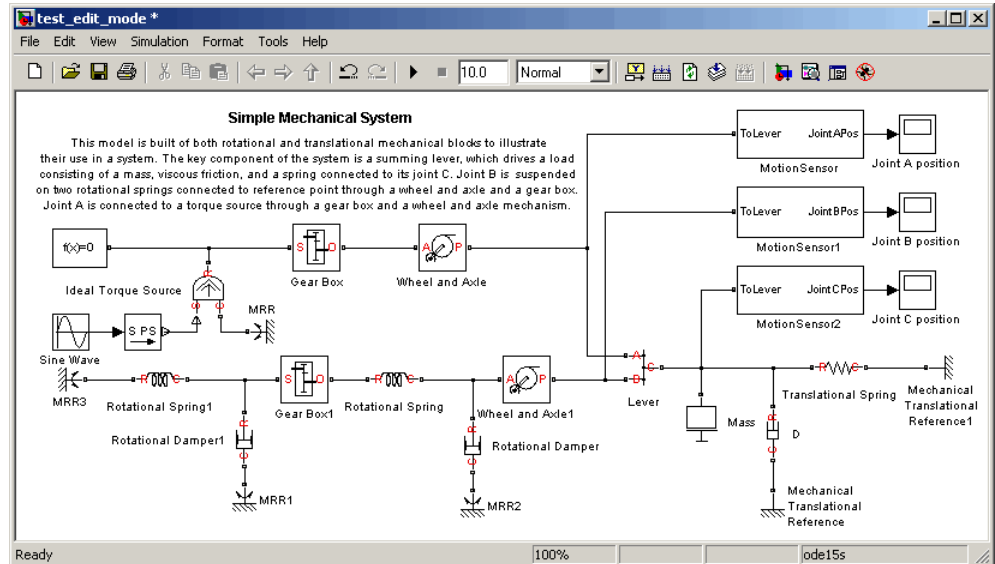
- 1 Open the test_edit_mode model, which you saved in Restricted mode in “Example of Saving a Model in Restricted Mode” on page 4-12. The model opens in Restricted mode.



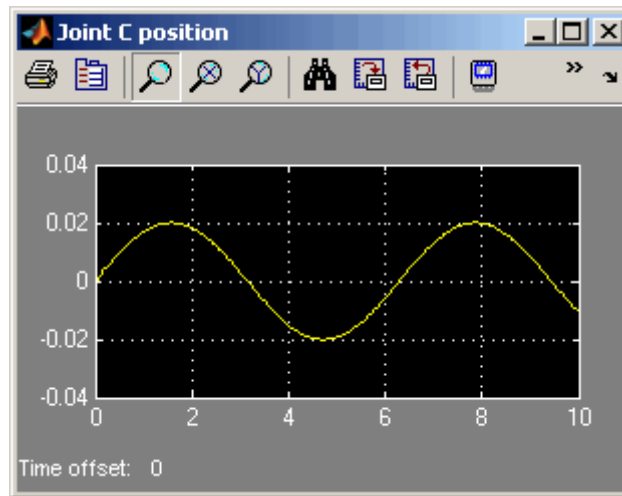
- 2 Open the Joint C Position scope and simulate the model.



- 3 Delete the Signal Builder block named Force Input. Replace it with a Sine Wave block from the Simulink Sources library, as shown below.



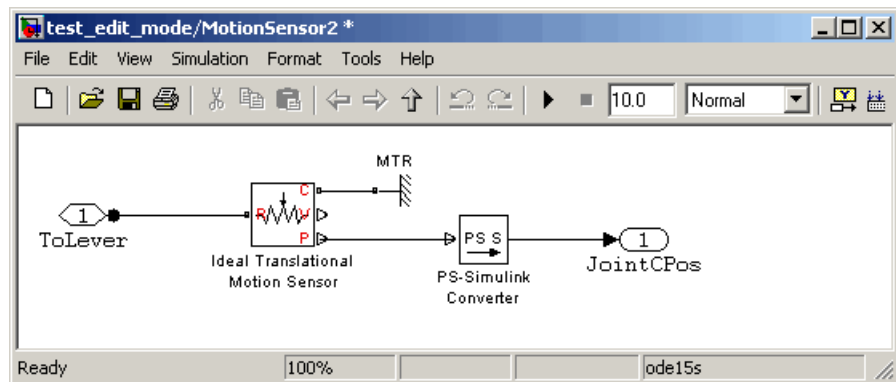
- 4 Simulate the model again. The model successfully compiles and simulates in Restricted mode.



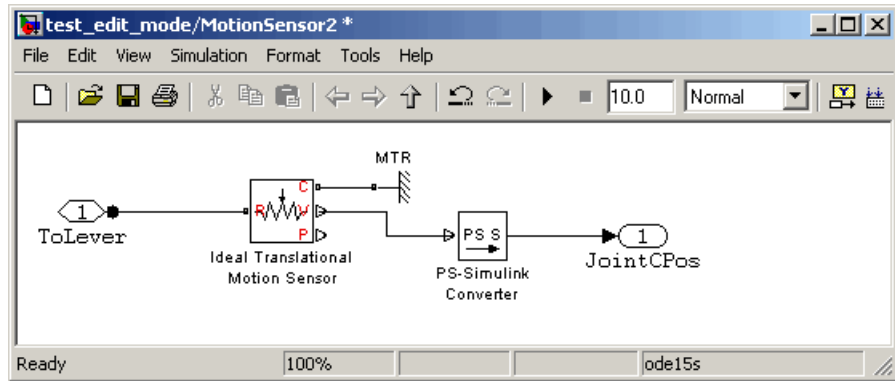
Performing an Operation Disallowed in Restricted Mode

This example shows what happens when you perform an operation that is disallowed in Restricted mode.

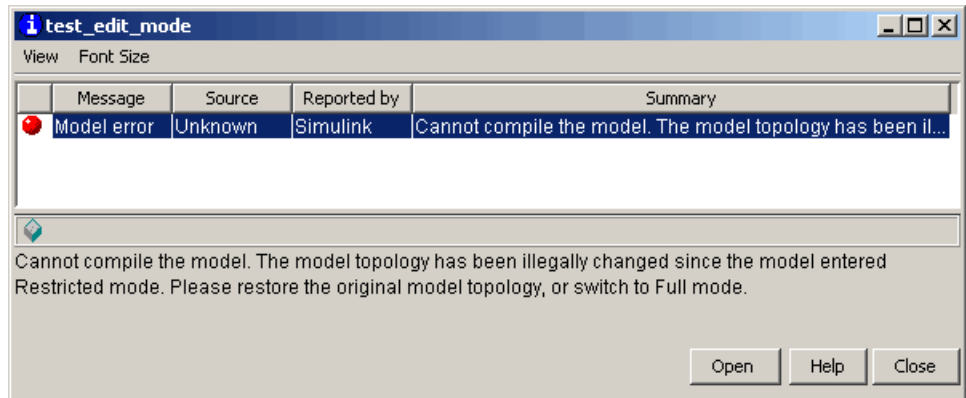
- 1 Open the `test_edit_mode` model, which you saved in Restricted mode in “Example of Saving a Model in Restricted Mode” on page 4-12. The model opens in Restricted mode.
- 2 Double-click the `MotionSensor2` block to open the subsystem.



- 3 Delete the connection line between port P of the Ideal Translational Motion Sensor block and the PS-Simulink Converter block. Instead, connect port V of the Ideal Translational Motion Sensor block to the input port of the PS-Simulink Converter block, to measure the velocity on node C of the lever.



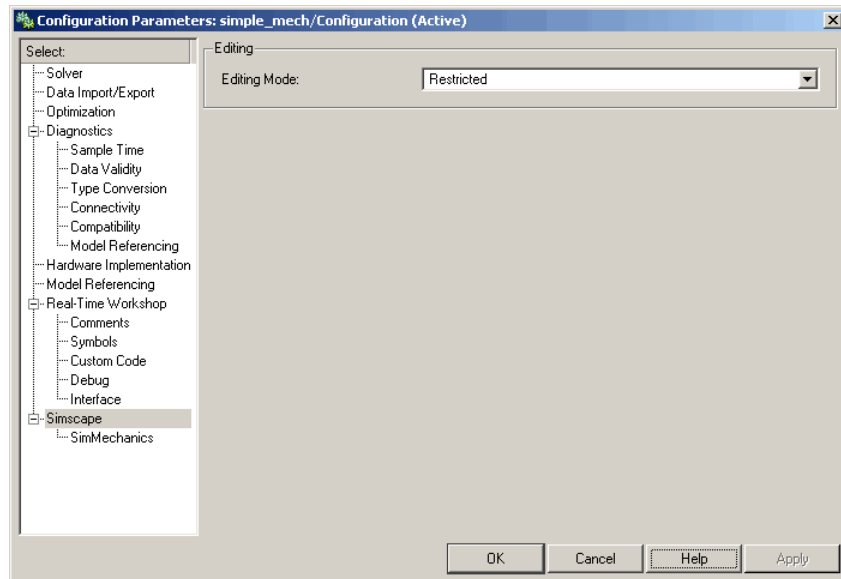
- 4 Try to simulate the model. An error message appears saying that the model cannot be compiled because its topology has been changed while in Restricted mode. You can either undo the changes, or switch to Full mode, as described in “Switching from Restricted to Full Mode” on page 4-20.



Switching from Restricted to Full Mode

If you need to perform a task that is disallowed in Restricted mode, you can try to switch the model to Full mode.

- 1 From the top menu bar in the model window, select **Simulation > Configuration Parameters**. The Configuration Parameters dialog box opens.
- 2 In the left pane of the Configuration Parameters dialog box, select **Simscape**. The right pane displays the **Editing Mode** option.
- 3 Select **Full** from the drop-down list, as shown, and click **OK**.



Simscape checks whether all the vertical product licenses for this model are available. If yes, it checks out the vertical product licenses and switches the model to Full mode. If a vertical product license is not available, Simscape issues an error message and the model stays in Restricted mode.

Note If the switch to Full mode fails but some of the vertical product licenses have already been checked out, they stay checked out until you quit the MATLAB session. For more information, see “Example with Multiple Vertical Products” on page 4-6.

Once the model is switched to Full mode, you can perform the needed design and simulation tasks, and then either save it in Full mode, or switch back to Restricted mode and save it in Restricted mode.

Editing Mode Information

- “What Is the Current Mode?” on page 4-23
- “Which Licenses Are Checked Out?” on page 4-23

What Is the Current Mode?

If you are unsure whether the model is currently open in Restricted or Full mode, you can check by following these steps.

- 1 From the top menu bar in the model window, select **Simulation > Configuration Parameters**. The Configuration Parameters dialog box opens.
- 2 In the left pane of the Configuration Parameters dialog box, select **Simscape**. The right pane displays the **Editing Mode** option, which is either Full or Restricted.
- 3 At this point, you can either try switching the mode by selecting a different option from the drop-down list, or click **Cancel** to stay in the current mode.

Which Licenses Are Checked Out?

Use the MATLAB `license` command to get a list of all the licenses currently in use. In the MATLAB Command Window, type

```
license('inuse')
```

This command returns a list of licenses checked out in the current MATLAB session. In the list, products are listed alphabetically by their license feature names.

across variables

Variables that are measured with a gauge connected in parallel to an element.

conserving ports

Bidirectional hydraulic or mechanical ports that represent physical connections and relate physical variables based on the Physical Network approach.

globally assigned positive direction

Direction considered positive for a model diagram.

nonrestricted parameters

Parameters that are available for modification when you open a model in Restricted mode. Usually, these are the block parameters with plain numerical values, such as **Chamber volume** or **Wheel radius**. Information on restricted and nonrestricted parameters is listed in block reference pages.

physical connections

Bidirectional connections between the blocks that mimic physical connections between elements.

physical signal ports

Unidirectional ports (inports and outports) transferring signals that use an internal Simscape engine for computations.

restricted parameters

Parameters that are not available for modification when you open a model in Restricted mode. You have to be in Full mode to modify them. Usually, these are the block parameterization options, such as **Chamber specification** or **Mechanism orientation**. Information on restricted and nonrestricted parameters is listed in block reference pages.

through variables

Variables that are measured with a gauge connected in series to an element.

vertical products

Products in the Physical Modeling family that use Simscape as a platform and, as a result, share common functionality such as physical units management, editing modes, and so on.

Examples

Use this list to find examples in the documentation.

Getting Started

“Creating a Simple Model” on page 2-7

Best Practices

“Example of Using a Parasitic Resistance to Avoid Numerical Simulation Issues” on page 2-26

Editing Mode

“Example of Saving a Model in Restricted Mode” on page 4-12

“Simulating and Fine-Tuning a Model in Restricted Mode” on page 4-13

“Adding and Deleting Simulink Blocks in Restricted Mode” on page 4-17

“Performing an Operation Disallowed in Restricted Mode” on page 4-19

N

numerical simulation issues
 avoiding 2-25

P

ports
 physical conserving 1-12
 physical signal 1-13

S

Simscape
 about 1-2
 block library structure 2-2
 editing modes 4-2
 vertical products 1-4
Simscape Editing Mode 4-2
 Full mode 4-3
 information 4-23
 Restricted mode 4-3
 saving in Restricted mode 4-10

switching between modes 4-4
workflows 4-2
working in Restricted mode 4-12
working with block libraries 4-7

T

terminology 4-1

U

units
 defining physical units 1-14

V

variables
 across 1-7
 direction 1-9
 through 1-7
 using in model equations 1-8