SimElectronics™ 1
User's Guide

MATLAB®
&SIMULINK®

The MathWorks™

Accelerating the pace of engineering and science
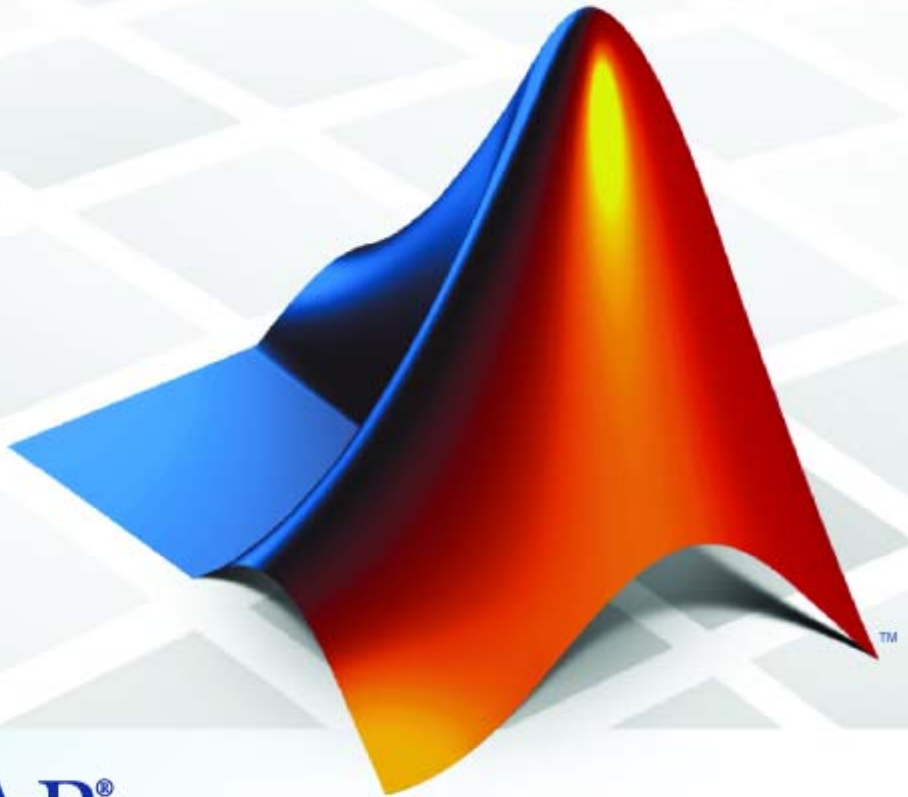
How to Contact The MathWorks

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

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

508-647-7000 (Phone)

508-647-7001 (Fax)

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

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

SimElectronics™ User's Guide

© COPYRIGHT 2008 by The MathWorks, Inc.

Trademarks

Patents

Revision History

April 2008	Online only	New for Version 1.0 (Release 2008a+)
October 2008	Online only	Revised for Version 1.1 (Release 2008b)

Contents

Getting Started

1

Modeling an Electronic System

2

Simulating an Electronic System

3

Examples

A

Index

1

Getting Started

Product Overview

In this section...
"Product Description" on page 1-2
"Assumptions and Limitations" on page 1-2
"Modeling Physical Networks with SimElectronics Blocks" on page 1-3

Product Description

SimElectronics™ software works with Simscape™ software and extends the physical modeling capabilities of the Simulink® product family with tools for modeling and simulating electromechanical and electronic systems. It contains blocks that let you model electromechanical and electronic systems at a speed and level of fidelity that is appropriate for system-level analysis. The blocks let you perform tradeoff analyses to optimize system design, for example, by testing various algorithms with different circuit implementations.

Assumptions and Limitations

Use SimElectronics with Simscape to simulate electromechanical and electrical systems in the time domain. It contains blocks that you can use with other Simscape blocks to simulate components at a system level. The library contains blocks that use either high-level or more detailed models to simulate components. SimElectronics does not have the capability to:

- Model large circuits with dozens of analog components, such as a complete transceiver.

- Perform either layout (physical design) tasks, or the associated implementation tasks such as layout versus schematic (LVS), design rule checking (DRC), parasitic extraction, and backannotation.

- Model 3-D parasitic effects that are typically important for high-frequency applications.

For these types of requirements, you must use an EDA package specifically designed for the implementation of analog circuits.

Another MathWorks™ product, SimPowerSystems™ software, is better suited for power system networks where:

- The underlying equations are predominantly linear (e.g., transmission lines and linear machine models).

- Three-phase motors and generators are used.

SimPowerSystems has blocks and solvers specifically designed for these types of applications.

Modeling Physical Networks with SimElectronics Blocks

SimElectronics is part of the Simulink Physical Modeling family. Models using SimElectronics are essentially Simscape block diagrams. To build a system-level model with electrical blocks, use a combination of SimElectronics blocks and other Simscape and Simulink blocks. You can connect SimElectronics blocks directly to Simscape blocks. You can connect Simulink blocks through the Simulink-PS Converter and PS-Simulink Converter blocks from the Simscape Utilities library. These blocks convert electrical signals to and from Simulink mathematical signals. For more information about connecting different types of blocks, see "Connecting Model Blocks" on page 2-6.

For more information about basic principles to follow when building an electrical model with SimElectronics, see "Basic Principles of Modeling Physical Networks" in the Simscape documentation.

Required and Related Products

Product Requirements

SimElectronics software is an extension of Simscape product, expanding its capabilities to model and simulate electronic and electromechanical elements and devices.

SimElectronics software requires these products:

- MATLAB®
- Simulink
- Simscape

Other Related Products

The SimElectronics product page at the MathWorks Web site lists the toolboxes and blocksets that extend the capabilities of MATLAB and Simulink. These products can enhance your use of SimElectronics software in various applications.

For more information about The MathWorks™ software products, see:

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

Product Demos

You can find interactive SimElectronics demos in the MATLAB Help browser:
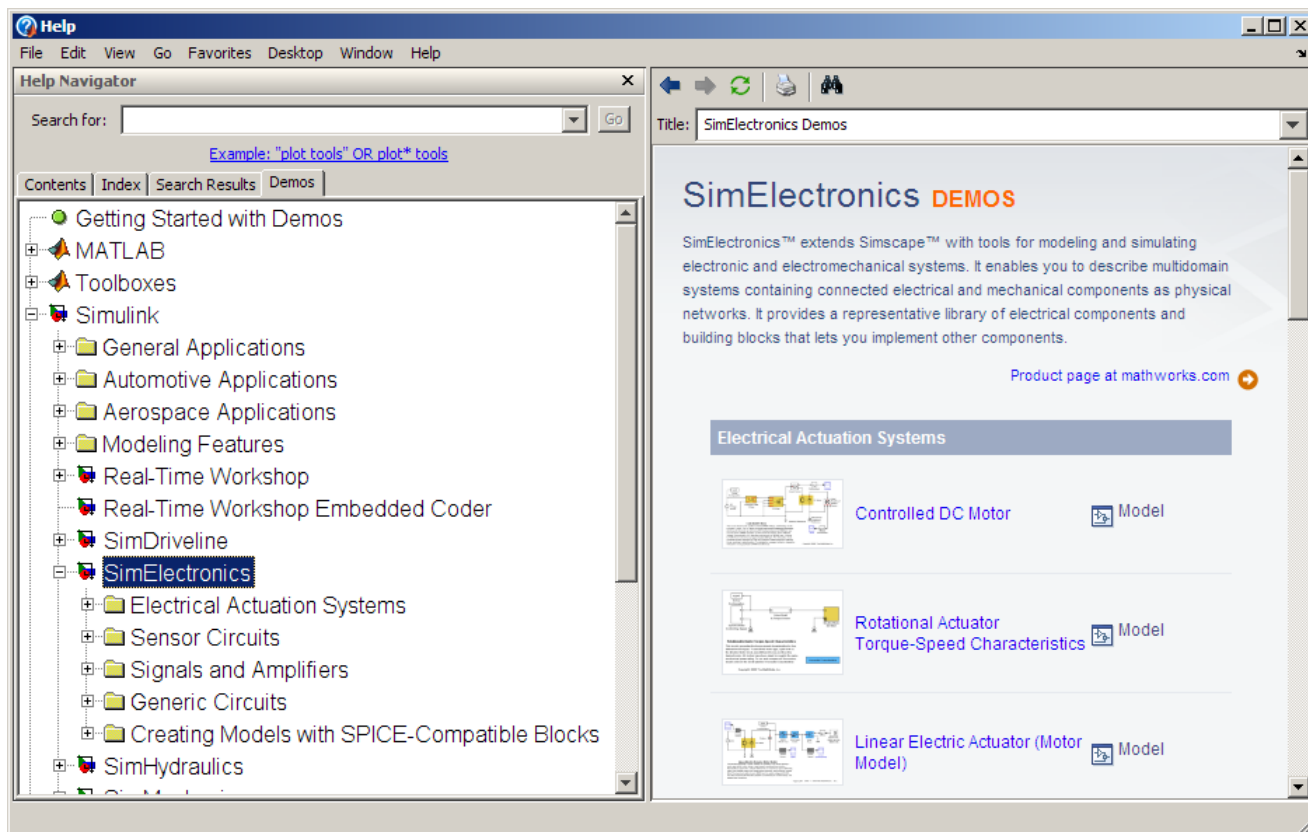
This example shows you how to locate and open an SimElectronics demo:

1 Type `demos` at the MATLAB prompt to open the Help browser to the
Demos tab.

2 Select **Simulink > SimElectronics** in the **Demos** tab to see a list of
demo categories.

3 Select a model, and click **Open this model** in the upper-right corner of the demo window to display the Simulink model for this demo.

4 In the model window, select **Simulation > Start** to run the demo simulation.

SimElectronics Block Libraries

In this section...

Overview of SimElectronics Libraries

SimElectronics libraries provide blocks for modeling electromechanical and electrical systems within the Simulink environment. SimElectronics also lets you create custom libraries to model circuit components defined in SPICE netlists.

At a high level, the library blocks fall into three categories:

- Electromechanical and electrical blocks — Blocks in the product that model electromechanical and electrical components using parameters that appear on manufacturer data sheets. These blocks provide multiple parameterizations when more than one datasheet parameterization convention is commonly used.

- SPICE-compatible blocks — Blocks in the product that model electrical components using industry-standard SPICE parameters.

- Custom blocks — Blocks you create that model a set of electrical components you define in a SPICE netlist.

Note SimElectronics follows the standard Simulink conventions where block inputs and outputs are called *ports*. In SimElectronics, each port represents a single electrical terminal.

A SimElectronics model can contain blocks from any of the standard SimElectronics libraries or from a custom library you create by importing circuit data from a SPICE netlist. A model can also include Simulink blocks and blocks from other products, such as those described in "Required and Related Products" on page 1-4.

This section contains the following topics:

Product Libraries

Use blocks from the SimElectronics libraries described in this table to model physical and electrical components by specifying physical and electrical properties.

Library	Description
Actuators & Drivers	Motors and drivers, specified using parameters that appear on manufacturer data sheets.
Integrated Circuits	Circuit-level components such as operational amplifiers.
Passive Devices	Resistive, inductive, and capacitive devices that depend on time or temperature.
Semiconductor Devices	Device-level components such as MOSFETs and bipolar transistors, including a sublibrary of SPICE-compatible blocks specified using industry-standard SPICE parameters.
Sensors	Electrical sensors, specified using parameters that appear on manufacturer data sheets.

Library	Description
Sources	Current and voltage sources, including a sublibrary of SPICE-compatible blocks specified using industry-standard SPICE parameters.
Utilities	Block for specifying simulation information that pertains to all SPICE-compatible blocks in a SimElectronics *network*, such as temperature. **Note** A SimElectronics network is a collection of one or more SimElectronics blocks. Collections of SimElectronics blocks that are separated by Simulink blocks (and the PS-Simulink Converter and Simulink-PS Converter blocks that bridge the SimElectronics and Simulink parts of the model) are in different networks.

For more information on defining components, see "Modeling Electronic Components" on page 2-2.

Custom SimElectronics Libraries

Use blocks from one or more custom libraries to model circuit components defined in SPICE netlists. You create the custom libraries by importing circuit information from a SPICE netlist into SimElectronics.

For more information on importing SPICE data, see "Importing SPICE Models Into a SimElectronics Library" on page 2-2.

Opening SimElectronics Libraries

There are two ways to access SimElectronics blocks:

- "Using the Simulink Library Browser to Access the Block Libraries" on page 1-10

- "Using the Command Prompt to Access the Block Libraries" on page 1-10

Using the Simulink Library Browser to Access the Block Libraries

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.

For more information on using the Library Browser, see "Library Browser" in the *Simulink Graphical User Interface* documentation.

Using the Command Prompt to Access the Block Libraries

Another way to access the block libraries is to open them individually by using the command prompt:

- To open just the SimElectronics library, type `elec_lib` in the MATLAB Command Window.

- To open the Simscape library (to access the utility blocks, as well as electrical sources, sensors, and other Foundation library blocks), 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 SimElectronics library window is shown in the following figure. Each icon in the window represents a library. Some of these libraries contain

second-level sublibraries. Double-click an icon to open the corresponding library.
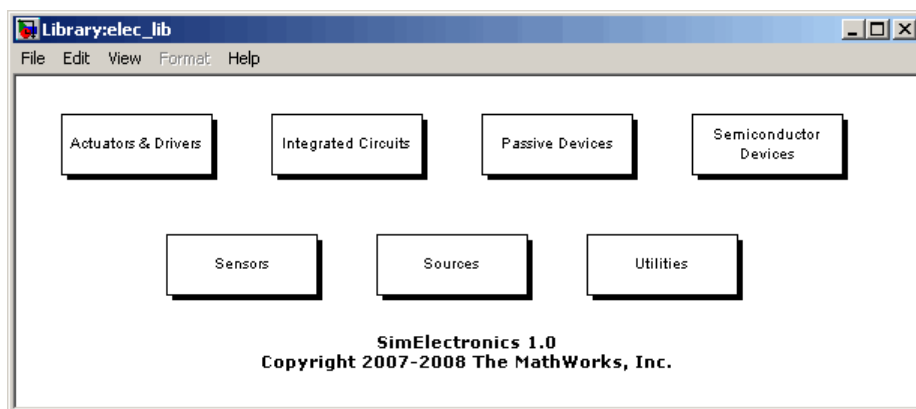
Product Workflow

When you analyze an electronic or electromechanical system using SimElectronics software, your workflow might include the following tasks:

1 Create a Simulink model that includes electronic or electromechanical components.

In the majority of applications, it is most natural to model the physical system using Simscape and SimElectronics blocks, and then develop the controller or signal processing algorithm in Simulink.

For more information about modeling the physical system, see "Modeling Electronic Components" on page 2-2.

2 Define component data by:

- Specifying electrical or mechanical properties
- Importing circuit data from industry-standard SPICE files

3 Configure the solver options.

For more information about the settings that most affect the solution of a physical system, see Chapter 3, "Simulating an Electronic System".

4 Run the simulation.

For more information on how the product performs time-domain simulation of an electronic system, see "Running a Time-Domain Simulation" on page 3-6.

Example — Modeling a DC Motor

In this section...
"Overview of DC Motor Example" on page 1-13
"Selecting Blocks to Represent System Components" on page 1-13
"Building the Model" on page 1-14
"Specifying Model Parameters" on page 1-16
"Configuring the Solver Parameters" on page 1-23
"Running the Simulation and Analyzing the Results" on page 1-24

Overview of DC Motor Example

In this example, you model a DC motor driven by a constant input signal that approximates a pulse-width modulated signal and look at the current and rotational motion at the motor output.

Selecting Blocks to Represent System Components

Select the blocks to represent the input signal, the DC motor, and the motor output displays.

The following table describes the role of the blocks that represent the system components.

Block	Description
Solver Configuration	Defines solver settings that apply to all physical modeling blocks.
DC Voltage Source	Generates a DC signal.
Controlled PWM Voltage	Generates the signal that approximates a pulse-width modulated motor input signal.
H-Bridge	Drives the DC motor.
Current Sensor	Converts the electrical current that drives the motor into a physical signal proportional to the current.

Block	Description
Ideal Rotational Motion Sensor	Converts the rotational motion of the motor into a physical signal proportional to the motion.
DC Motor	Converts input electrical signal into mechanical motion.
PS-Simulink Converter	Converts the input physical signal to a Simulink signal.
Scope	Displays motor current and rotational motion.
Electrical Reference	Provides the electrical ground.
Mechanical Rotational Reference	Provides the mechanical ground.

Building the Model

Create a Simulink model, add blocks to the model, and connect the blocks.

1 Create a model.

If you are new to Simulink, see the "Creating a Simulink Model" example for information on how to create a model.

2 Add to the model the blocks listed in the following table. The Library column of the table specifies the hierarchical path to each block.

Block	Library Path	Quantity
Solver Configuration	**Simscape > Utilities**	1
DC Voltage Source	**Simscape > Foundation Library > Electrical > Electrical Sources**	1
Controlled PWM Voltage	**Simscape > SimElectronics > Actuators & Drivers**	1

Block	Library Path	Quantity
H-Bridge	Simscape > SimElectronics > Actuators & Drivers	1
Current Sensor	Simscape > Foundation Library > Electrical > Electrical Sensors	1
Ideal Rotational Motion Sensor	Simscape > Foundation Library > Mechanical > Mechanical Sensors and Sources	1
DC Motor	Simscape > SimElectronics > Actuators & Drivers	1
PS-Simulink Converter	Simscape > Utilities	2
Scope	Simulink > Commonly Used Blocks	2
Electrical Reference	Simscape > Foundation Library > Electrical > Electrical Elements	1
Mechanical Rotational Reference	Simscape > Foundation Library > Mechanical > Rotational Elements	1

3 Connect the blocks as shown in the following figure.
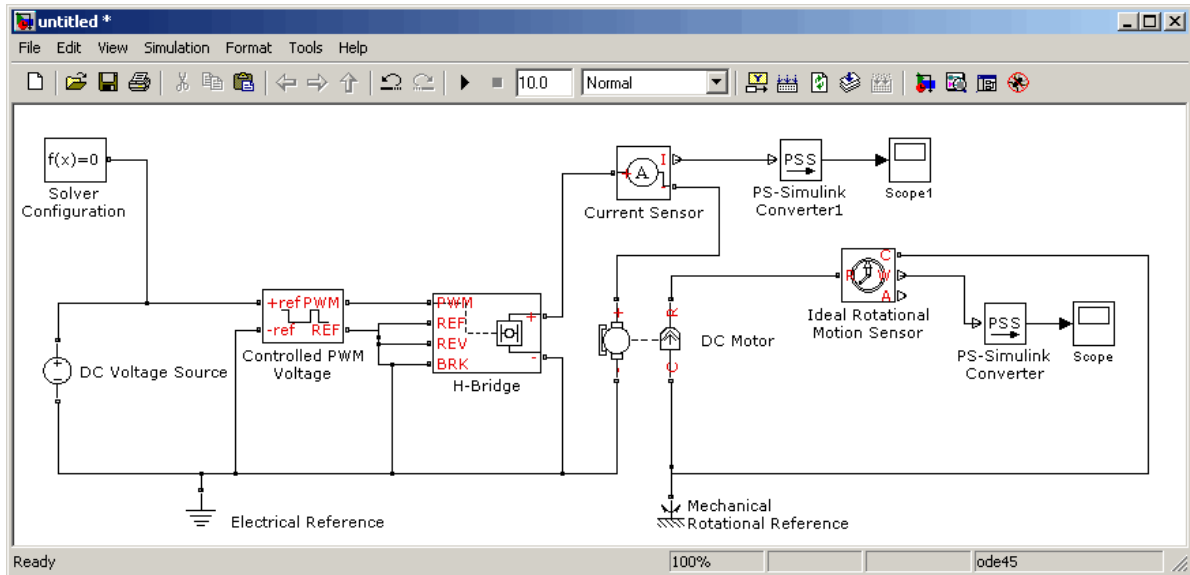
Now you are ready to specify block parameters.

Specifying Model Parameters

Specify the following parameters to represent the behavior of the system components:

- "Model Setup Parameters" on page 1-16
- "Motor Input Signal Parameters" on page 1-17
- "Motor Parameters" on page 1-21
- "Current Display Parameters" on page 1-21
- "Torque Display Parameters" on page 1-22

Model Setup Parameters

The following blocks specify model information that is not specific to a particular block:

- Solver Configuration
- Electrical Reference
- Mechanical Rotational Reference

As with Simscape models, you must include a Solver Configuration block in each topologically distinct physical network. This example has a single physical network, so use one Solver Configuration block with the default parameter values.

You must include an Electrical Reference block in each SimElectronics network. You must include a Mechanical Rotational Reference block in each network that includes electromechanical blocks. These blocks do not have any parameters.

For more information about using reference blocks, see "Grounding Rules" in the Simscape documentation.

Motor Input Signal Parameters

You generate the motor input signal using three blocks:

- The DC Voltage Source block generates a constant signal.
- The Controlled PWM Voltage block generates a pulse-width modulated signal.
- The H-Bridge block drives the motor.

In this example, all input ports of the H-Bridge block except the PWM port are connected to ground. As a result, the H-Bridge block behaves as follows:

- When the motor is on, the H-Bridge block connects the motor terminals to the power supply.
- When the motor is off, the H-Bridge block acts as a freewheeling diode to maintain the motor current.

In this example, you simulate the motor with a constant current whose value is the average value of the PWM signal. By using this type of signal, you set up a fast simulation that estimates the motor behavior.

1 Set the DC Voltage Source block parameters as follows:

- **Constant voltage** = 2.5
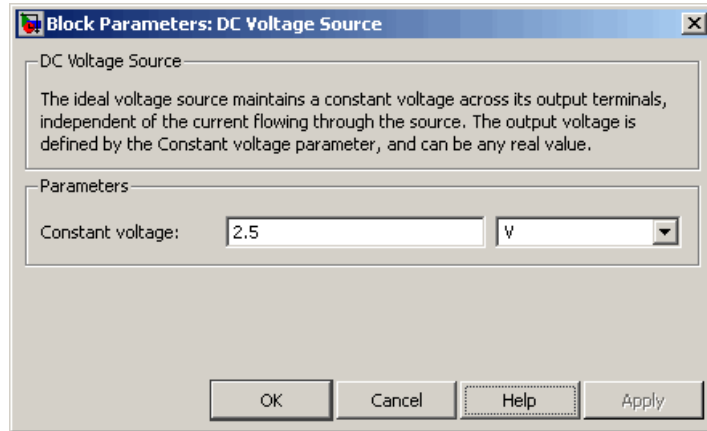
2 Set the Controlled PWM Voltage block parameters as follows:

- **PWM frequency** = 4000

- **Simulation mode** = Averaged

  This value tells the block to generate an output signal whose value is the average value of the PWM signal. Simulating the motor with an averaged signal estimates the motor behavior in the presence of a PWM signal. To validate this approximation, use value of PWM for this parameter.
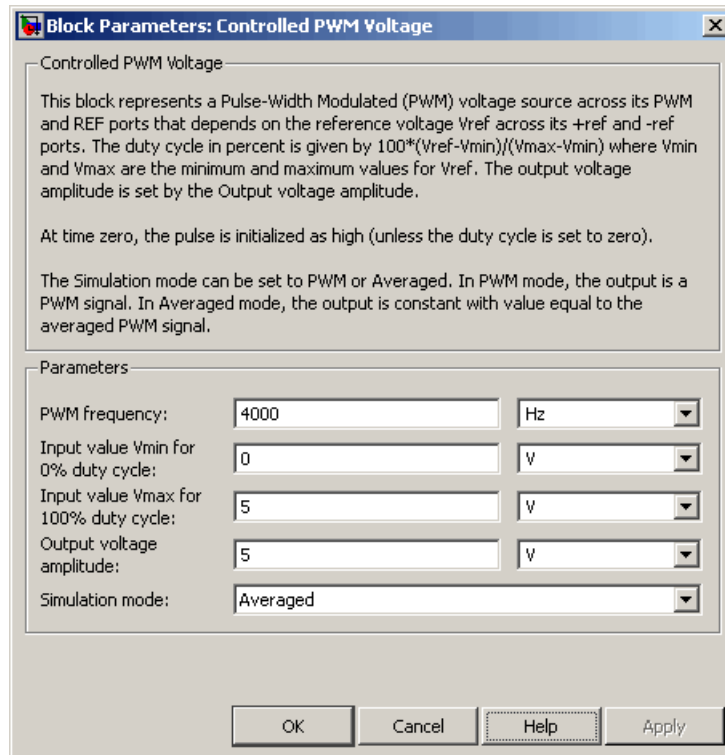
3 Set the H-Bridge block parameters as follows:

- **Simulation mode** = Averaged

  This value tells the block to generate an output signal whose value is the average value of the PWM signal. Simulating the motor with an averaged signal estimates the motor behavior in the presence of a PWM signal. To validate this approximation, use value of PWM for this parameter to validate this approximation.
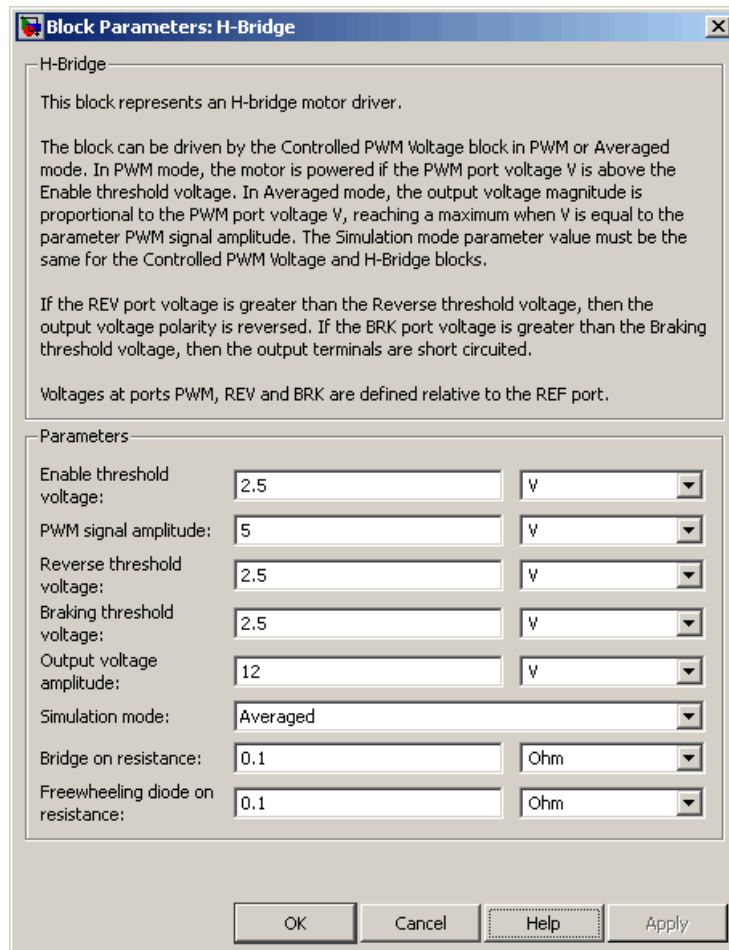
Motor Parameters

Configure the block that models the motor.

Set the Motor block parameters as follows, leaving the unit settings at their default values where applicable:

- **Electrical Torque** tab:
  - **Model parameterization** = By rated power, rated speed & no-load speed
  - **Armature inductance** = 0.01
  - **No-load speed** = 4000
  - **Rated speed (at rated load)** = 2500
  - **Rated load (mechanical power)** = 10
  - **Rated DC supply voltage** = 12
- **Mechanical** tab:
  - **Rotor inertia** = 2000
  - **Rotor damping** = 1e-06

Current Display Parameters

Specify the parameters of the blocks that create the motor current display:

- Current Sensor block
- PS-Simulink Converter1 block
- Scope1 block

Of the three blocks, only the PS-Simulink Converter1 block has parameters. Set the PS-Simulink Converter1 block **Output signal unit** parameter to A to indicate that the block input signal has units of amperes.
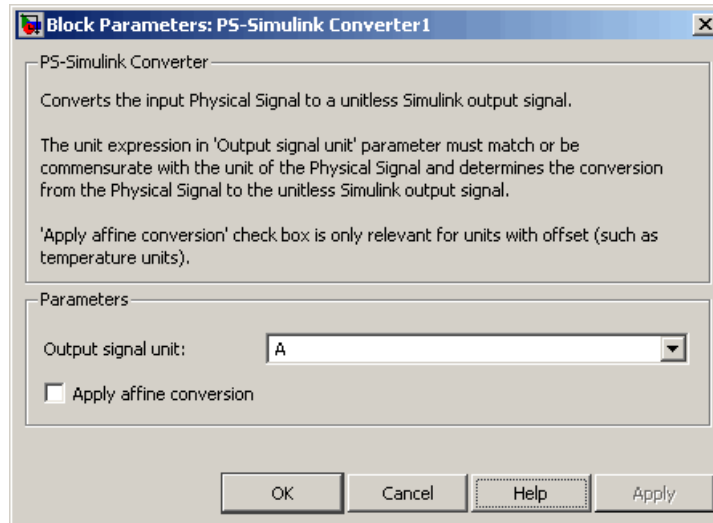
Torque Display Parameters

Specify the parameters of the blocks that create the motor torque display:

- Ideal Rotational Motion Sensor block
- PS-Simulink Converter block
- Scope block

Of the three blocks, only the PS-Simulink Converter block has parameters you need to configure for this example. Set the PS-Simulink Converter block **Output signal unit** parameter to rpm to indicate that the block input signal has units of revolutions per minute.

Note You must type this parameter value. It is not available in the drop-down list.

Block Parameters: PS-Simulink Converter

PS-Simulink Converter

Converts the input Physical Signal to a unitless Simulink output signal.

The unit expression in 'Output signal unit' parameter must match or be commensurate with the unit of the Physical Signal and determines the conversion from the Physical Signal to the unitless Simulink output signal.

'Apply affine conversion' check box is only relevant for units with offset (such as temperature units).

Parameters

Output signal unit: [rpm]

☐ Apply affine conversion

[OK]   [Cancel]   [Help]   [Apply]

Configuring the Solver Parameters

Configure the solver parameters to use a continuous-time solver because SimElectronics models only run with a continuous-time solver. Increase the maximum step size the solver can take so the simulation runs faster.

1 In the model window, select **Simulation > Configuration Parameters** to open the **Configuration Parameters** dialog box.

2 Select ode15s (Stiff/NDF) from the **Solver** list.

3 Enter 1 for the **Max step size** parameter value.

4 Click **OK**.
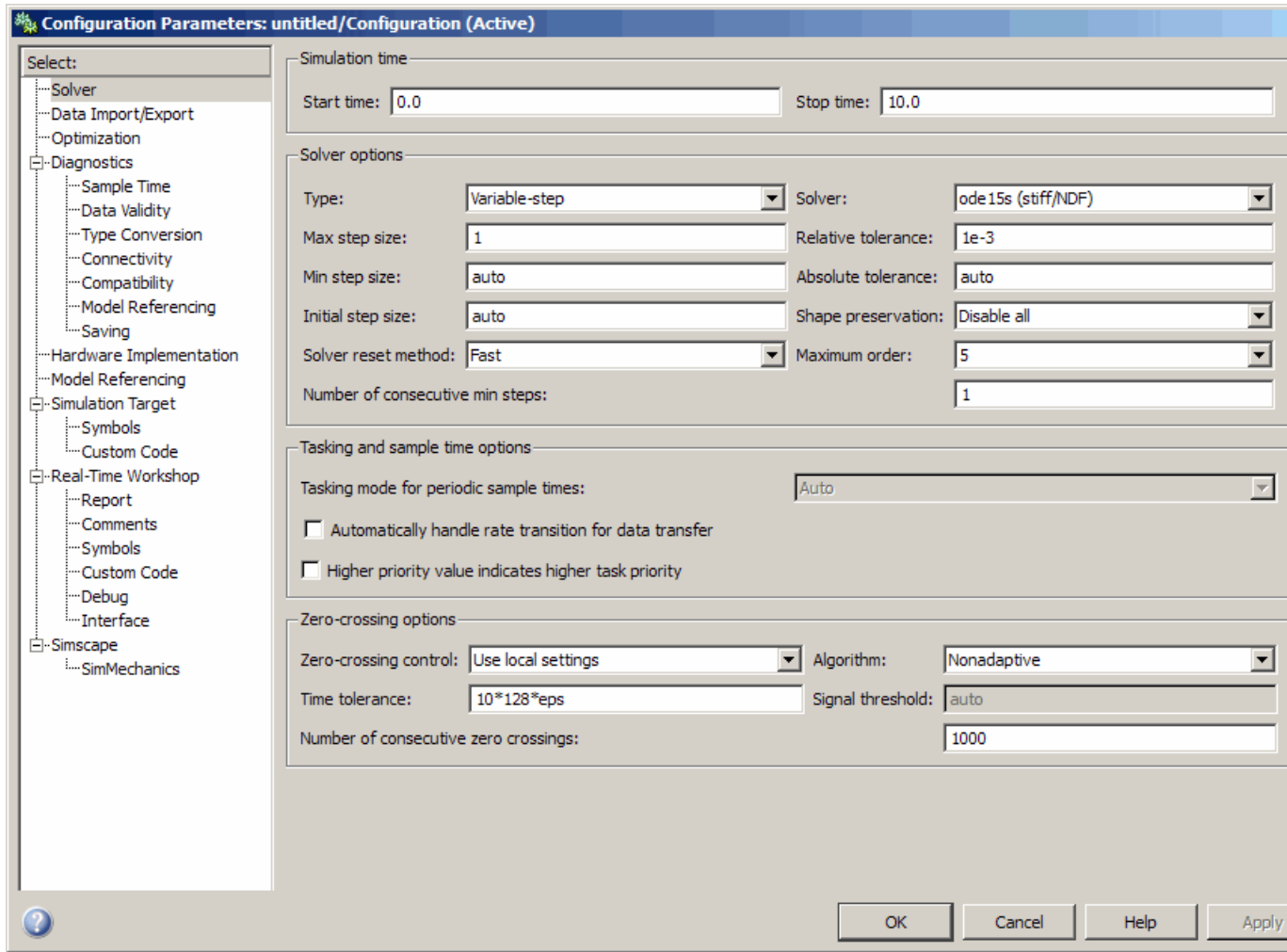
For more information about configuring solver parameters, see Chapter 3, "Simulating an Electronic System".

Running the Simulation and Analyzing the Results

In this part of the example, you run the simulation and plot the results.

In the model window, select **Simulation > Start** to run the simulation.

To view the motor current and torque in the Scope windows, double-click the Scope blocks. You can do this before or after you run the simulation.

Note By default, the scope displays appear stacked on top of each other on the screen, so you can only see one of them. Click and drag the windows to reposition them.

The following plot shows the motor current.
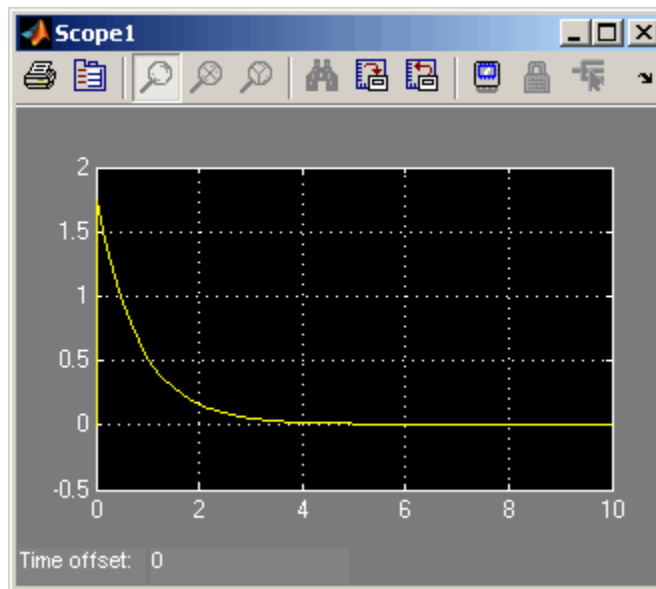
Motor Current

The next plot shows the motor rpm.
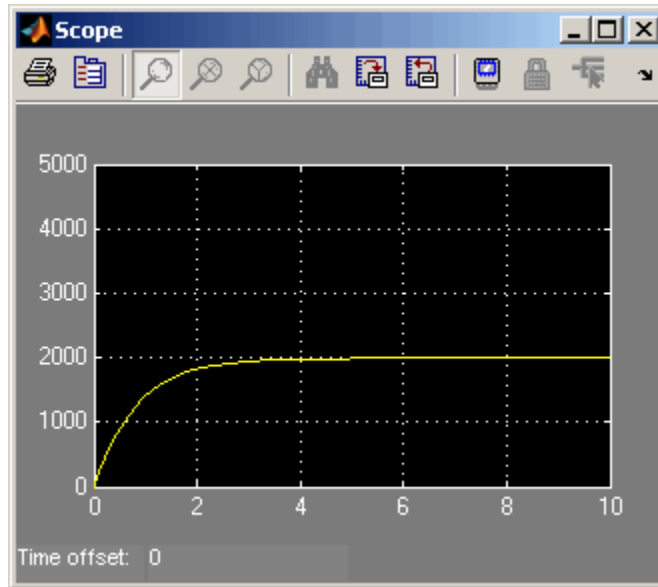
Motor RPM

As expected, the motor runs at about 2000 rpm when the applied DC voltage is 2.5 V.

Example — Modeling a Triangle Wave Generator

In this section...
"Overview of Triangle Wave Generator Example" on page 1-27
"Selecting Blocks to Represent System Components" on page 1-27
"Creating the Blocks that Represent SPICE Netlists" on page 1-29
"Building the Model" on page 1-30
"Specifying Model Parameters" on page 1-33
"Configuring the Solver Parameters" on page 1-39
"Running the Simulation and Analyzing the Results" on page 1-39

Overview of Triangle Wave Generator Example

In this example, you model a triangle wave generator you build from
SimElectronics electrical blocks and custom SimElectronics blocks that
represent SPICE netlists, and then look at the voltage at the wave generator
output.

You use a classic circuit configuration consisting of an integrator and a
noninverting amplifier to generate the triangle wave.

Selecting Blocks to Represent System Components

First, you select the blocks to represent the input signal, the triangle wave
generator, and the output signal display.

You model the triangle wave generator with a set of physical blocks bracketed
by a Simulink-PS Converter block and a PS-Simulink Converter block. The
wave generator consists of:

- Two operational amplifier blocks
- Two diode blocks
- Resistors and a capacitor that work with the operational amplifiers to
  create the integrator and noninverting amplifier

- Simulink-PS Converter and PS-Simulink Converter blocks. The function of the Simulink-PS Converter and PS-Simulink Converter blocks is to bridge the physical part of the model, which uses bidirectional physical signals, and the rest of the model, which uses unidirectional Simulink signals.

You have descriptions of the diodes and operational amplifiers you want to model in SPICE netlist form. Later in the example, you use a SimElectronics function to create a block that represents each of these components.

Note When you create a block that represents a component from a SPICE netlist, the block name is the subcircuit name that appears in the netlist.

The following table describes the role of the blocks that represent the system components.

Block	Description
Sine Wave	Generates a sinusoidal signal that controls the resistance of the Variable Resistor block.
Simulink-PS Converter	Converts the sinusoidal Simulink signal to a physical signal.
SPICE Environment Parameters	Defines parameters that apply to all SPICE-compatible physical modeling blocks.
Solver Configuration	Defines solver settings that apply to all physical modeling blocks.
Electrical Reference	Provides the electrical ground.
Capacitor	Works with an operational amplifier and resistor block to create the integrator.
Resistor	Works with the operational amplifier and capacitor blocks to create the integrator and noninverting amplifier.
Variable Resistor	Supplies a time-varying resistance that adjusts the gain of the integrator, which in turn varies the frequency and amplitude of the generated triangular wave.

Block	Description
DC Voltage Source	Generates a DC reference signal for the operational amplifier block of the noninverting amplifier.
Voltage Sensor	Converts the electrical voltage at the output of the integrator into a physical signal proportional to the current.
PS-Simulink Converter	Converts the output physical signal to a Simulink signal.
Scope	Displays the triangular output wave.
OPAMP	Works with the capacitor and resistor to create an integrator and a noninverting amplifier.
dzrect	Limits the amplitude of the noninverting amplifier output.

Creating the Blocks that Represent SPICE Netlists

Create the blocks that represent the operational amplifier and diode described in the SPICE files OpAmp.cir and PartsLib.cir, respectively.

1 Type the following at the MATLAB prompt to create the OPAMP library block:

```
netlist2sl('OpAmpCir.cir', 'genlib')
```

This command tells SimElectronics to map all circuit data in the file OpAmpCir.cir to a library called genlib. The file defines a single subcircuit called OPAMP, so genlib contains a single block called OPAMP after you issue the command.

Note When you import subcircuit data from a SPICE netlist, SimElectronics creates a block that represents the netlist, but you cannot look at the underlying circuit in the Simulink environment. You can look at the netlist by clicking **Help** in the block dialog box.

2 Type the following at the MATLAB prompt to create the dzrect library block:

```
options.LibName = 'genlib'
options.ModelOnly = 1
netlist2sl('PartsLib.cir', options)
```

These commands tell the SimElectronics software to map only the model card data in the file PartsLib.cir to a library called genlib. The file defines a single model card called dzrect, so these commands add a single block by this name to genlib.

Note When you import model card data from a SPICE netlist, SimElectronics uses the data to populate the parameter values of the corresponding SPICE-compatible SimElectronics block. In this example, dzrect is a Diode block from the SPICE-Compatible Semiconductors library whose parameter values come from the diode model card defined in PartsLib.cir.

For more information about creating SimElectronics blocks that represent circuit data in SPICE netlists, see "Importing SPICE Models Into a SimElectronics Library" on page 2-2 or the netlist2sl reference page.

Building the Model

Create a Simulink model, add blocks to the model, and connect the blocks.

1 Create a model.

If you are new to Simulink, see the "Creating a Simulink Model" example for information on how to create a model.

2 Add to the model the blocks listed in the following table. The Library column of the table specifies the hierarchical path to each block.

Block	Library Path	Quantity
Sine Wave	Simulink > Sources	1

Block	Library Path	Quantity
Simulink-PS Converter	**Simscape > Utilities**	1
SPICE Environment Parameters	**Simscape > SimElectronics > Utilities**	1
Solver Configuration	**Simscape > Utilities**	1
Electrical Reference	**Simscape > Foundation Library > Electrical > Electrical Elements**	1
Capacitor	**Simscape > Foundation Library > Electrical > Electrical Elements**	1
Resistor	**Simscape > Foundation Library > Electrical > Electrical Elements**	3
Variable Resistor	**Simscape > Foundation Library > Electrical > Electrical Elements**	1
DC Voltage Source	**Simscape > Foundation Library > Electrical > Electrical Sources**	1
Voltage Sensor	**Simscape > Foundation Library > Electrical > Electrical Sensors**	1
PS-Simulink Converter	**Simscape > Utilities**	1
Scope	**Simulink > Commonly Used Blocks**	1
OPAMP	`genlib`	2
dzrect	`genlib`	2

3 Connect the blocks as shown in the following figure.

Note Use **Ctrl+R** to rotate the Resistor, Capacitor, and dzrect1 blocks so the polarity matches that shown in the figure.
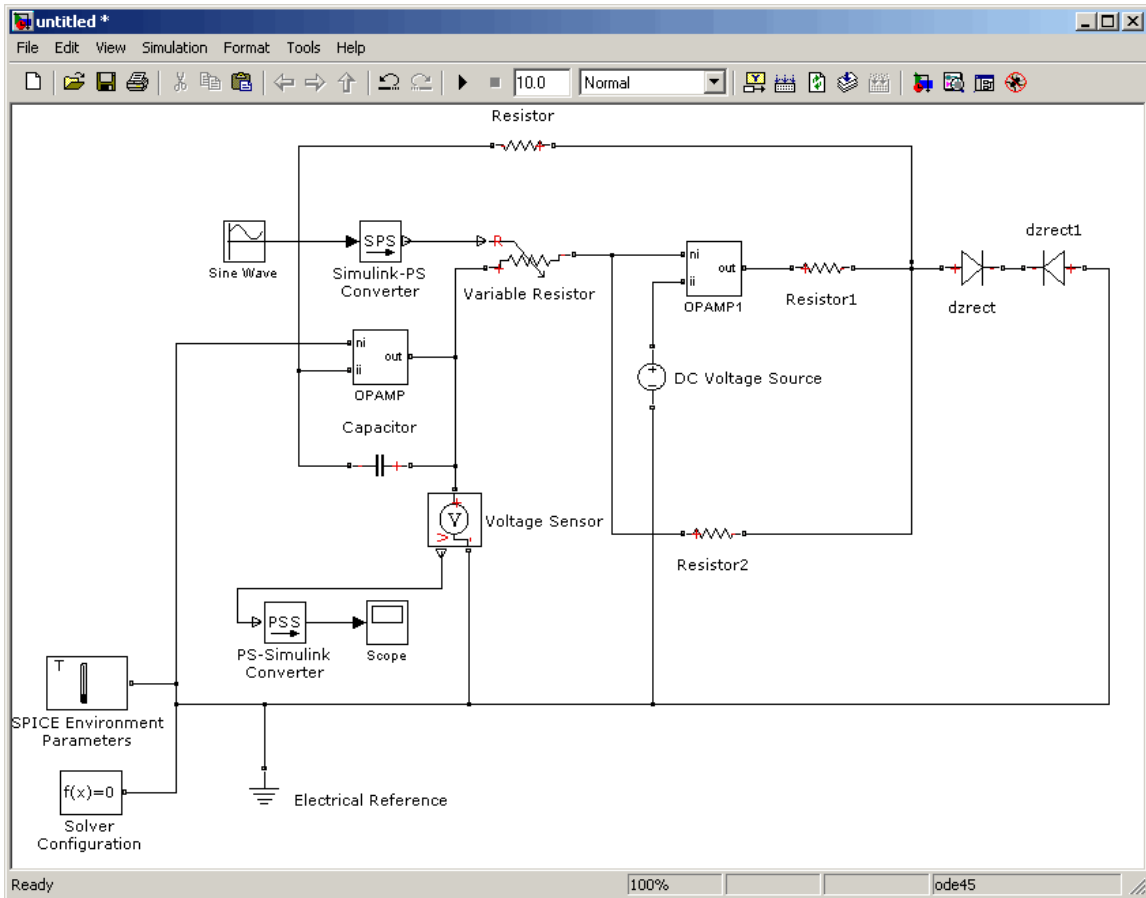
Now you are ready to specify block parameters.

Specifying Model Parameters

Specify the following parameters to represent the behavior of the system components:

- "Model Setup Parameters" on page 1-33
- "Input Signal Parameters" on page 1-33
- "Triangle Wave Generator Parameters" on page 1-34
- "Signal Display Parameters" on page 1-39

Model Setup Parameters

The following blocks specify model information that is not specific to a particular block:

- Solver Configuration
- Electrical Reference
- SPICE Environment Parameters

As with Simscape models, you must include a Solver Configuration block in each topologically distinct physical network. This example has a single physical network, so use one Solver Configuration block with the default parameter values.

You must include an Electrical Reference block in each SimElectronics network. This block does not have any parameters.

The SPICE Environment Parameters block establishes temperature and minimum conductance values for the SPICE-compatible blocks. You must include a SPICE Environment Parameters block in each SimElectronics network to adjust these values. In this example, you use the default parameter values, so the block is only included in the model to show you how to use it.

Input Signal Parameters

Generate the sinusoidal control signal using the Sine Wave block.

Set the Sine Wave block parameters as follows:

- **Amplitude** = `0.5e4`

- **Bias** = `1e4`

- **Frequency (rad/sec)** = `pi/5e-4`
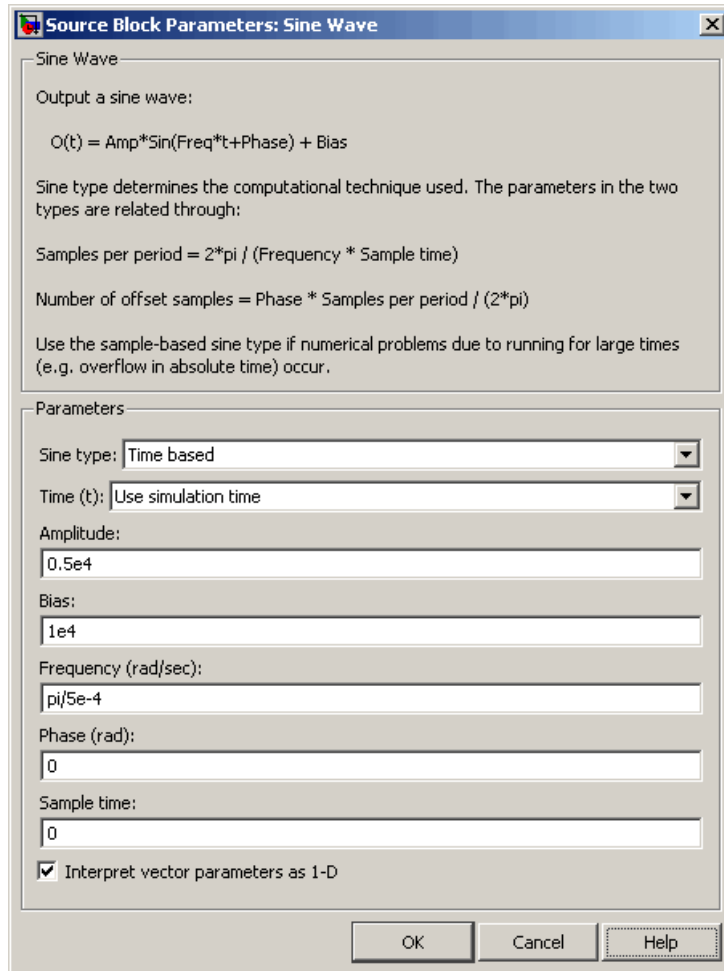
Triangle Wave Generator Parameters

Configure the blocks that model the physical system that generates the triangle wave:

- Integrator — OPAMP, Capacitor, and Resistor blocks
- Noninverting amplifier — OPAMP1, Resistor2, and Variable Resistor blocks
- Resistor1
- Simulink-PS Converter and PS-Simulink Converter blocks that bridge the physical part of the model and the Simulink part of the model.

1 Accept the default parameters for the Simulink-PS Converter block. These parameters establish the units of the physical signal at the block output such that they match the expected default units of the Variable Resistor block input.

2 The OPAMP and OPAMP1 blocks do not have parameters because you created these blocks from a SPICE subcircuit.

3 Accept the default parameter values for the dzrect and dzrect1 blocks because these are the values you imported from the SPICE model card.

4 The Voltage Sensor block does not have any parameters.

5 Accept the default parameters for the Variable Resistor block. These parameters establish the units of the physical signal at the block output such that they match the expected default units of the Variable Resistor block input.

6 Set the Capacitor block parameters as follows:

- **Capacitance** = `2.5e-9`
- **Initial voltage** = `0.08`

    This value starts the oscillation in the feedback loop.

- **Series resistance** = `0`
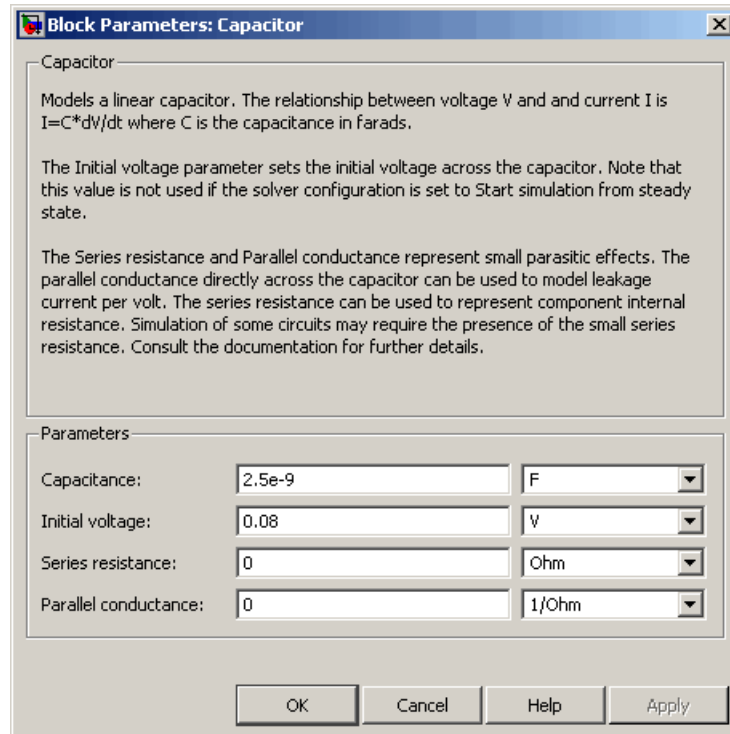
7 Set the DC Voltage Source block parameters as follows:

- **Constant voltage** = 0
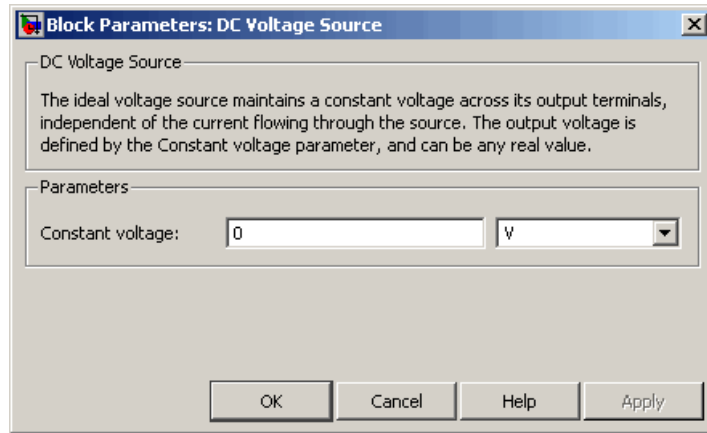
8 Set the Resistor block parameters as follows:
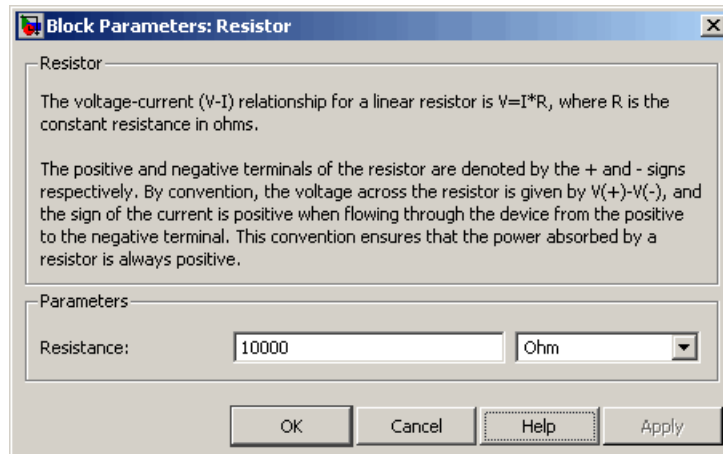
- **Resistance** = 10000
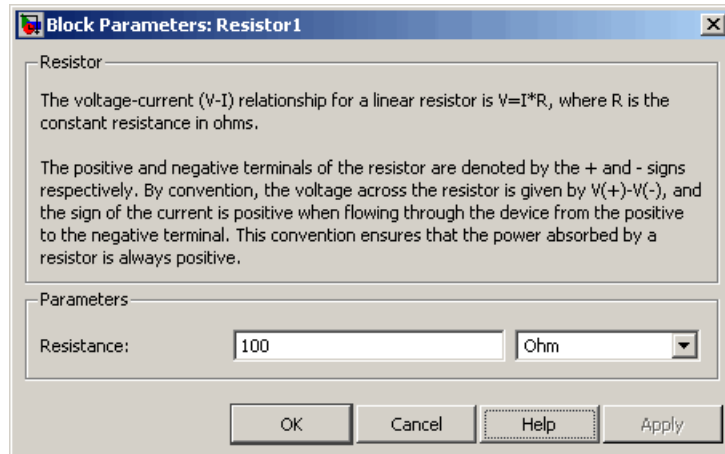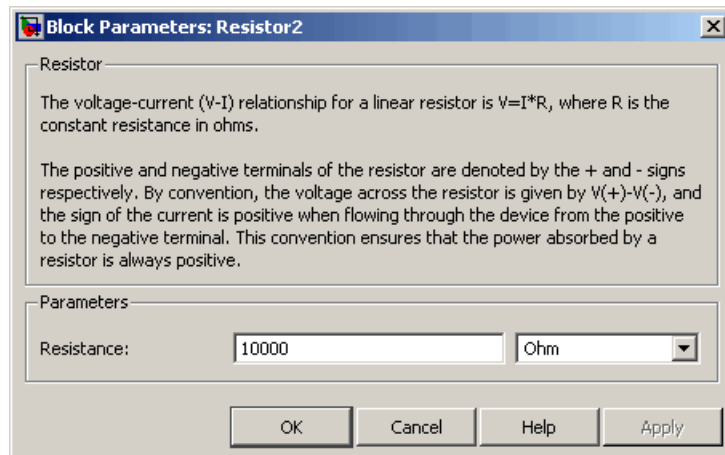

9 Set the Resistor1 block parameters as follows:

- **Resistance** = 100

10 Set the Resistor2 block parameters as follows:

- **Resistance** = 10000

11 Accept the default parameters for the PS-Simulink Converter block. These parameters establish the units of the physical signal at the block output such that they match the expected default units of the Scope block input.

Signal Display Parameters

Specify the parameters of the Scope block to display the triangular output signal.

Double-click the Scope block and then double-click the **Parameters** button

to open the Scope parameters dialog box. On the **Data history** tab, clear the **Limit data points to last** check box.

Configuring the Solver Parameters

Configure the solver parameters to use a continuous-time solver because SimElectronics models only run with a continuous-time solver. You also change the simulation end time, tighten the relative tolerance for a more accurate simulation, and remove the limit on the number of simulation data points Simulink saves.

1 In the model window, select **Simulation > Configuration Parameters** to open the **Configuration Parameters** dialog box.

2 In the **Solver** category in the **Select** tree on the left side of the dialog box:

- Enter `2000e-6` for the **Stop time** parameter value.

- Select `ode23t (Mod.  stiff/Trapezoidal)` from the **Solver** list.

- Enter `4e-5` for the **Max step size** parameter value.

- Enter `1e-6` for the **Relative tolerance** parameter value.

3 In the **Data Import/Export** category in the **Select** tree:

- Clear the **Limit data points to last** check box.

4 Click **OK**.

For more information about configuring solver parameters, see Chapter 3, "Simulating an Electronic System".

Running the Simulation and Analyzing the Results

Run the simulation and plot the results.

In the model window, select **Simulation > Start** to run the simulation.

To view the triangle wave in the Scope window, double-click the Scope block. You can do this before or after you run the simulation.

The following plot shows the voltage waveform. As the resistance of the Variable Resistor block increases, the amplitude of the output waveform increases and the frequency decreases.
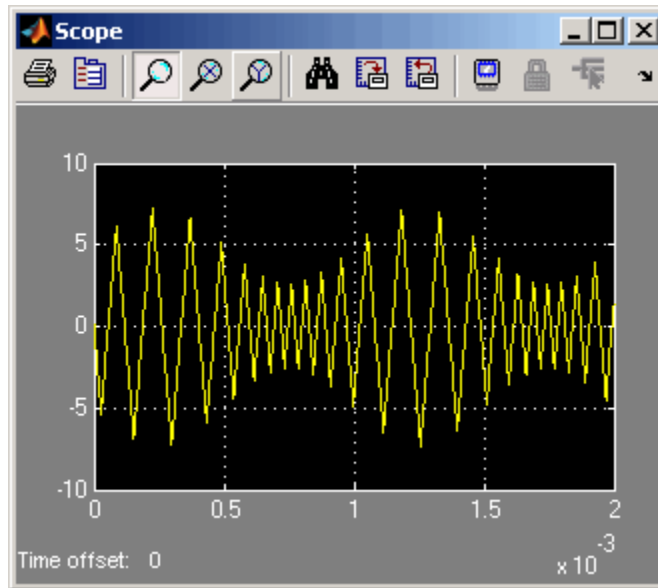
Triangle Waveform Voltage

Modeling an Electronic System

Modeling Electronic Components

In this section...
"Importing SPICE Models Into a SimElectronics Library" on page 2-2
"Adding SimElectronics Blocks to a Model" on page 2-5
"Connecting Model Blocks" on page 2-6

Importing SPICE Models Into a SimElectronics Library

You can import circuit data from a SPICE netlist into a SimElectronics library. This lets you simulate SPICE circuits that describe the following types of devices within the Simulink environment:

- Resistors, capacitors, and inductors
- Voltage and current sources
- Diodes
- Bipolar transistors
- JFETs
- MOSFETs

This section contains the following topics:

- "Supported SPICE Netlists" on page 2-2
- "How to Import SPICE Models" on page 2-4

Supported SPICE Netlists

You can only import SPICE netlist files that define one or more SPICE subcircuits or model cards. The subcircuits can contain other subcircuits and model cards.

SimElectronics software lets you import:

- All subcircuit data in the netlist file. SimElectronics maps the SPICE netlist to a library that contains one or more blocks, with one block representing each subcircuit in the file.

  When you import subcircuit data from a SPICE netlist, SimElectronics creates a *black-box model* of the circuit, that is, SimElectronics creates a block that represents the netlist, but you cannot look at the underlying circuit in the Simulink environment.

- Only the model card data in the netlist file. SimElectronics maps the SPICE netlist to a library that contains one or more blocks, with one block representing each model card in the file.

  When you import model card data from a SPICE netlist, SimElectronics uses the data to populate the parameter values of the corresponding SPICE-compatible SimElectronics block. You can verify the imported values by opening the block.

Note SimElectronics cannot model large circuits with dozens of analog components. If you try to simulate a model with more than a few tens of components, you may encounter numerical simulation issues.

The following table lists the device models you can import into SimElectronics.

Device Type	SPICE Device Model(s)
DC Source	DC
Exponential Source	EXP
Voltage- and Current-Controlled Sources with one controlling voltage or current	E, F, G, H
Sinusoidal Source	SIN
Piecewise Linear Source	PWL
Pulse Source	PULSE
Single-Frequency FM	SFFM
Resistor	R

Device Type	SPICE Device Model(s)
Capacitor	C
Inductor	L
Diode	D
Bipolar Transistor	NPN, PNP
JFET	NJF, PJF
MOSFET	NMOS1, PMOS1

When you import circuit data from a SPICE netlist, SimElectronics ignores parameters that pertain to AC analysis because the software only simulates the resulting blocks in the time domain.

SimElectronics does not support SPICE noise parameters.

How to Import SPICE Models

Use the `netlist2sl` function to import information from a SPICE netlist into Simulink. This function maps the SPICE netlist to a SimElectronics library that contains one or more blocks.

The following command illustrates how to create a library called `libraryname` from the subcircuit in the netlist file `filename`. The library contains a block whose name matches the name of the subcircuit and has the same inputs and outputs.

```
netlist2sl(filename, libraryname)
```

The `netlist2sl` reference page describes the function arguments in detail and also explains how to use the function to import only the model card information from the netlist into a Simulink library.

Note If the library already exists in the specified directory, `netlist2sl` adds new blocks to this library. If a subcircuit name conflicts with an existing block name in the library, `netlist2sl` prompts you to either overwrite the existing block or rename the new block.

For an example of importing a SPICE model into a custom library, see either "Example — Modeling a Triangle Wave Generator" on page 1-27 or the SimElectronics demo, Creating Custom Library Blocks from Circuit Netlist.

Adding SimElectronics Blocks to a Model

You can include blocks from the SimElectronics library in a Simulink model. For more information on the library and the SimElectronics blocks, see "SimElectronics Block Libraries" on page 1-7.

This section contains the following topics:

- "Required Blocks" on page 2-5
- "How to Add SimElectronics Blocks to a Model" on page 2-5

Required Blocks

Each topologically distinct physical network in a diagram requires exactly one Solver Configuration block, found in the Simscape Utilities library. The Solver Configuration block specifies global environment information for simulation and provides parameters for the solver that your model needs before you can begin simulation. For more information, see the Solver Configuration block reference page.

Each electrical network requires an Electrical Reference block. This block establishes the electrical ground for the circuit. Networks with electromechanical blocks also require a Mechanical Rotational Reference block. For more information about using reference blocks, see "Grounding Rules" in the Simscape documentation.

How to Add SimElectronics Blocks to a Model

To add SimElectronics blocks to a Simulink model:

1 Type `elec_lib` at the MATLAB prompt to open the SimElectronics library.

2 Navigate to the desired library or sublibrary.

3 Drag instances of SimElectronics blocks into the model window using the mouse.

> **Note** You can also access SimElectronics blocks and other Simulink blocks from the Simulink Library Browser window. Type `simulink` at the MATLAB prompt to open this window. Add blocks to the model by dragging them from this window and dropping them into the model window.

Connecting Model Blocks

You follow the same procedure for connecting SimElectronics blocks as for connecting Simulink blocks: click a port and drag the mouse to draw a line to another port on a different block. For more information on connecting blocks, see "Connecting Blocks in the Model Window" in the Simulink documentation.

You can only connect blocks that use the same type of signal. SimElectronics blocks use the same physical signals that Simscape blocks use. These signals are different than Simulink signals, and are represented graphically by a different port style. Therefore, you can freely connect pairs of SimElectronics blocks and other Simscape blocks.

However, you cannot directly connect SimElectronics blocks to Simulink blocks. Instead, you must use the Simscape PS-Simulink Converter and Simulink-PS Converter blocks to bridge them.

The following topics in the Simscape documentation explain how physical ports work and how to bridge physical blocks and Simulink blocks.

- "Connector Ports and Connection Lines"
- "Connecting Simscape Diagrams to Simulink Sources and Scopes"

Specifying Parameters that Affect All SPICE-Compatible Blocks

Block that Controls Network-Level Parameter Values

The SimElectronics SPICE Environment Parameters block lets you set the following *network-level* parameter values, that is, values that apply to all SPICE-compatible blocks in an electrical network:

- Circuit temperature

- Minimum conductance

Note Only SPICE-compatible blocks use the temperature setting in the SPICE Environment Parameters block. Other blocks are not temperature dependent. Instead, SimElectronics simulates these blocks at the **Measurement temperature** parameter value you specify in the corresponding block dialog box. SimElectronics uses this parameter value to establish the temperature-independent block behavior.

Specifying Network-Level Parameter Values

To specify a circuit operating temperature or minimum conductance for the SPICE-compatible blocks, add a SPICE Environment Parameters block to the circuit and adjust one or more of these parameters:

- **Circuit temperature**

- **Minimum conductance GMIN**

For more information about these parameters, see the SPICE Environment Parameters block reference page.

Working with Simulink Blocks

| In this section... |
| --- |
| "Modeling Instantaneous Events" on page 2-9 |
| "Using Simulink Blocks to Model Physical Components" on page 2-9 |

Modeling Instantaneous Events

When working with SimElectronics software, your model may include Simulink blocks that create instantaneous changes to the physical system inputs through the Simulink-PS Converter block, such as those associated with events or discrete sampling. When you build this type of model, make sure the corresponding zero crossings are generated.

Many blocks in the Simulink library generate these zero crossings by default. For example, the Pulse Generator block produces a discrete-time output by default, and generates the corresponding zero crossings. To model instantaneous events, select Use local settings or Enable all for the **Zero crossing control** option under the model's Solver Configuration Parameters to generate zero crossings. For more information about zero crossing control, see "Zero-crossing control" in the Simulink documentation.

Using Simulink Blocks to Model Physical Components

To run a fast simulation that approximates the behavior of the physical components in a system, you may want to use Simulink blocks to model of one or more physical components.

The Modeling an Integrated Circuit demo uses Simulink to model a physical component. The Simulink Logical Operator block implements the behavioral model of the two-input NOR gate, as shown in the following figure.
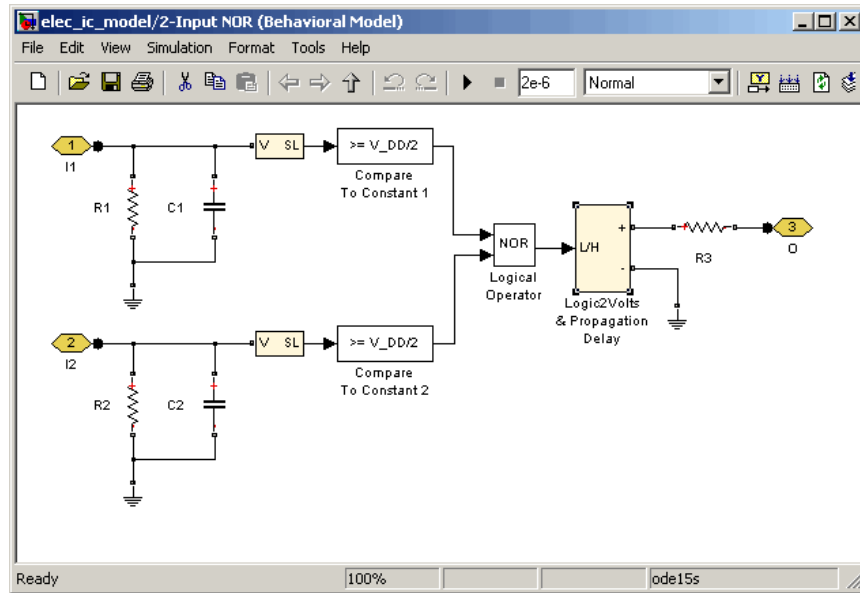
Using Simulink in this manner introduces algebraic loops, unless you place a lag somewhere between the physical signal inputs and outputs. In this case, a first-order lag is included in the Logic2Volts & Propagation Delay subsystem to represent the delay due to gate capacitances. For applications where no lag is required, use blocks from the Physical Signals sublibrary in the Simscape Foundation Library to implement the desired functionality.

Simulating an Electronic System

Selecting a Solver

In this section...
"Available Solvers" on page 3-2
"How to Select a Solver" on page 3-2

Available Solvers

SimElectronics software supports all of the continuous-time solvers that Simscape supports:

- ode23t — Trapezoidal solver

- ode15s — Variable order solver

- ode14x — Fixed-step solver

You can select any of the supported solvers for running a SimElectronics simulation. The variable-step solvers, ode23t and ode15s, are recommended for most applications because they run faster and copy better for systems with a range of both fast and slow dynamics. The ode23t solver is closest to the solver that SPICE traditionally uses.

To use Real-Time Workshop® software to generate standalone C or C++ code from your model, you must use the ode14x solver. For more information about code generation, see "Generating Code" in the Simscape documentation.

How to Select a Solver

To specify a solver:

1 Open the **Configuration Parameters** dialog box from the **Simulation** menu.

2 In the **Solver** pane (which opens by default), select the desired solver from the **Solver** list.

Specifying Simulation Accuracy/Speed Tradeoff

In this section...
"Parameters that Affect Accuracy and Speed" on page 3-3
"Determining Appropriate Accuracy/Speed Parameter Values" on page 3-3

Parameters that Affect Accuracy and Speed

To trade off accuracy and simulation time, adjust one or more of the following parameters:

- **Relative tolerance** (in the Configuration Parameters block dialog box)
- **Absolute tolerance** (in the Configuration Parameters block dialog box)
- **Max step size** (in the Configuration Parameters block dialog box)
- **Constraint Residual Tolerance** (in the Solver Configuration block dialog box)

Determining Appropriate Accuracy/Speed Parameter Values

In most cases, the default tolerance values produce accurate results without sacrificing unnecessary simulation time. The parameter value that is most likely to be inappropriate for your simulation is **Max step size**, because the default value, auto, depends on the simulation start and stop times rather than on the amount by which the signals are changing during the simulation. If you are concerned about the solver's missing significant behavior, change the parameter to prevent the solver from taking too large a step.

The Simulink documentation describes the following parameters in more detail and provides tips on how to adjust them:

- "Relative tolerance"
- "Absolute tolerance"
- "Max Step Size"

The Solver Configuration block reference page in the Simscape documentation explains when to adjust the **Constraint Residual Tolerance** parameter value.

Avoiding Simulation Issues

In this section...

General Troubleshooting for Simscape Models

If you experience a simulation issue, first read "Troubleshooting Simulation Errors" in the Simscape documentation to learn about general troubleshooting techniques.

Note As mentioned in the "Product Overview" on page 1-2, SimElectronics software does not have the ability to model large circuits with dozens of analog components. If you encounter convergence issues when trying to simulate a model with more than a few tens of transistors, you may find that the limitations of SimElectronics software prevent you from achieving convergence with any set of simulation parameter values.

Troubleshooting for Simscape Models that Include SimElectronics Blocks

There are a few techniques you can apply to any SimElectronics model to overcome simulation issues:

- Add parasitic capacitors and/or resistors (specifically, junction capacitance and ohmic resistance) to the circuit to avoid numerical issues. The Astable Oscillator demo uses these devices.

- Adjust the current and voltage sources so they start at zero and ramp up to their final values rather than starting at nonzero values.

"Working with Simulink Blocks" on page 2-9 describes how to avoid simulation errors in the presence of specific SimElectronics model configurations.

Running a Time-Domain Simulation

When you run a time-domain simulation, SimElectronics software uses the Simscape solver to analyze the physical system in the Simulink environment. For more information about how Simscape simulation works, see "How Simscape Simulation Works" in the Simscape documentation.

Running a Small-Signal Frequency-Domain Analysis

In this section...

Linearizing SimElectronics Models

The Simulink commands `linmod` and `dlinmod` 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.

For more information about linearizing models that contain blocks from the Simulink Physical Modeling family, see "Linearizing Simscape Models".

After you linearize your model, you can perform small-signal analyses such as:

- Bode

- Step

- Impulse

- Nyquist

Note If you have Simulink® Control Design™ software installed, you can use its Control and Estimation Tools Manager GUI to linearize models. For more information about using this software for linearization, see "Linearizing Models in the Control and Estimation Tools Manager".

Analyzing Small-Signal Behavior Using Bode Plots

After you create an LTI state-space model from your SimElectronics model, you can create a Bode plot for small-signal analysis using one of the following approaches:

- "Using MATLAB for Bode Analysis" on page 3-8

Using MATLAB for Bode Analysis

You can write a MATLAB script to generate a bode plot of the frequency response of the model from the LTI state-space model. The demo Small-Signal Frequency-Domain Analysis uses a linear passive bandpass filter example to show how to useMATLAB to create a Bode plot from an LTI model. The demo shows how to use both the state-space and transfer-function forms of the LTI model.

Using Control System Toolbox Software for Bode Analysis

You can use the built-in analysis and plotting capabilities of Control System Toolbox™ to understand the behavior of a linearized model.

- The toolbox function `ss` converts the state-space model returned by `linmod` to a continuous-time state-space model.

- The toolbox function `tf` converts the transfer function model returned by `linmod` to a continuous-time transfer function model.

- The toolbox function `ltiview` opens the LTI viewer for LTI system response analysis.

The following example shows how to create a Bode plot of the frequency-response of the model in the demo Small-Signal Frequency-Domain Analysis.

At the MATLAB prompt:

1 Type the following to create a LTI state-space model of `elec_ss_analysis`:

```
[A B C D] = linmod('elec_ss_analysis',[],[],[1e-5 0 1]);
```

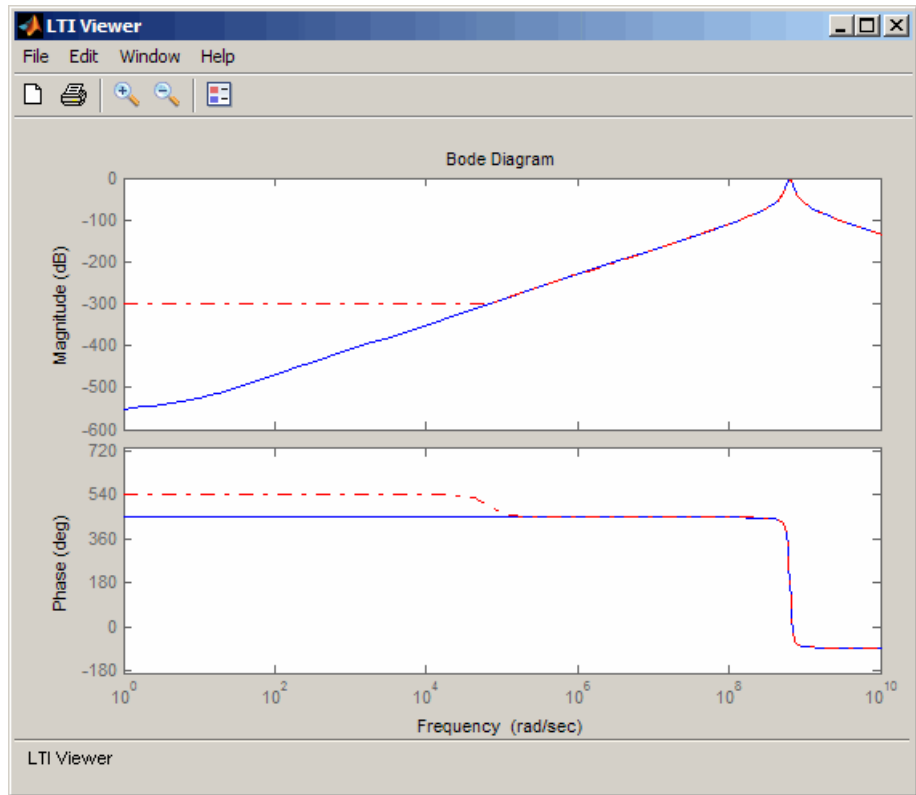**2** Type the following to create a LTI transfer function model of `elec_ss_analysis`:

```
[num den] = linmod('elec_ss_analysis',[],[],[1e-5 0 1]);
```

3 Type the following to plot both the state-space and transfer function models on a Bode plot:

```
ltiview('bode',ss(A,B,C,D),'b',tf(num,den),'r-.');
```

The toolbox creates the following plot:

The bandpass filter frequency response for the state-space and transfer function representations are different far from the center frequency. These differences arise from numerical noise introduced when calculating the transfer function representation. `linmod` implicitly derives a state-space representation for a linearized model, and computes the transfer function representation from this state-space model.

Note If you have Simulink Control Design software installed, you can use its Control and Estimation Tools Manager GUI to analyze models. This software uses the Control System Toolbox LTI viewer to display and analyze the dynamic behavior of a model.

Examples

Use this list to find examples in the documentation.

Examples

Index