Simscape™ Power Systems™ Release Notes
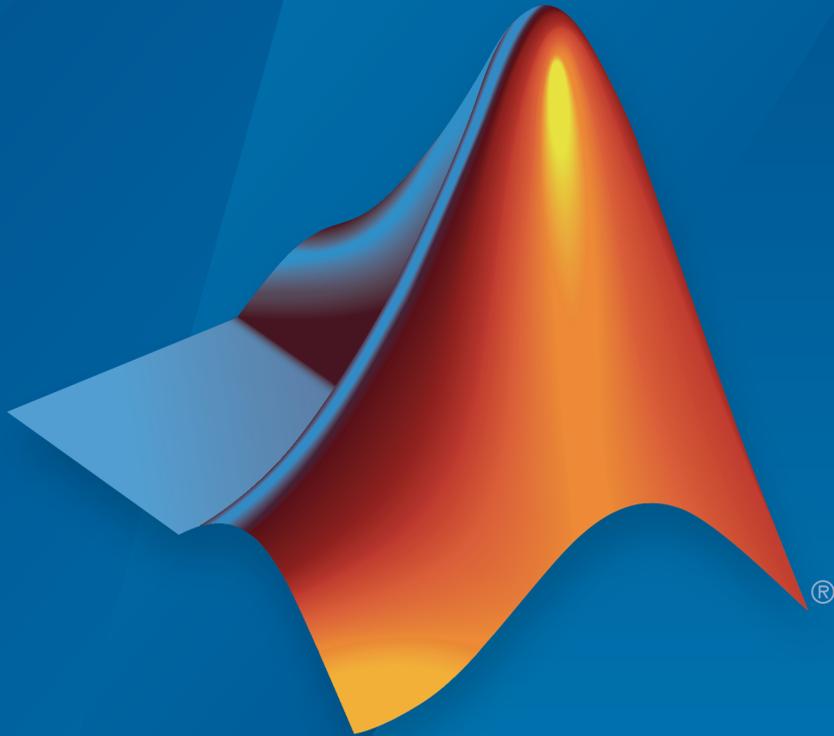
MATLAB®&SIMULINK®

MathWorks®

How to Contact MathWorks

	Latest news:	www.mathworks.com
	Sales and services:	www.mathworks.com/sales_and_services
	User community:	www.mathworks.com/matlabcentral
	Technical support:	www.mathworks.com/support/contact_us
	Phone:	508-647-7000

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

Contents

R2016b

R2016a

R2015b

R2015a

R2014b

R2014a

R2013b

R2013a

R2012b

R2012a

R2011b

R2011a

R2010b

R2010a

R2009b

R2007b

R2007a

R2006b

R2006a

R14SP3

Bug Fixes

R14SP2+

Bug Fixes

R2017a

Version: 6.7

New Features

Bug Fixes

Compatibility Considerations

Simscape Components

Battery Block Aging and Thermal Modeling: Simulate effects that degrade battery capacity and performance

A fade-modeling enhancement lets you analyze the effect of a long simulation, with multiple charge-discharge cycles, or define a starting point for a simulation based on the previous charge-discharge history. For example, if a vehicle has a new battery or a battery that requires replacing, you can simulate the electrical system of the vehicle without modifying multiple model parameters. To specify the dependence of other battery parameters on the charge-discharge history, set parameter **Model battery fade?** to `Include`.

The Battery block also has new modeling variants that allow you to:

- Simulate heat transfer.
- Access internal charge measurement.

To select a variant, right-click the block in your block diagram and then select the appropriate option from the context menu, under **Simscape** > **Block choices**:

- **Uninstrumented | No thermal port** — Basic model that does not output battery charge level or simulate thermal effects. This variant is the default.
- **Uninstrumented | Show thermal port** — Model with exposed thermal port. This model does not measure internal charge level of the battery.
- **Instrumented | No thermal port** — Model with exposed charge output port. This model does not simulate thermal effects.
- **Instrumented | Show thermal port** — Model that lets you measure internal charge level of the battery and simulate thermal effects. Both the thermal port and the charge output port are exposed.

The instrumented variants have an extra physical signal port that outputs the internal state of charge. Use this functionality to change the load behavior as a function of state of charge, without the complexity of building a charge state estimator.

The thermal port variants expose a thermal port, which represents the battery thermal mass. When you select this option, provide additional parameters to define battery behavior at a second temperature.

Caution: When using a thermal variant for a battery that operates at temperatures outside the temperature range bounded by the **Measurement temperature** and **Second temperature measurement** values, use caution. The block uses linear interpolation for the derived equation coefficients, and simulation results might become nonphysical outside the specified range. The block checks that the internal series resistance, self-discharge resistance, and nominal voltage remain positive, and issues error messages if there is a violation.

When you open a model that contains a Battery block from a previous release, the block is configured as the **Uninstrumented | No thermal port** variant. The parameters and variables map according to the table. For more information on configuring the block parameters and variables, see Battery.

R2017a	Previous to R2017a	Value	Notes
Nominal voltage	**Nominal voltage, Vnom**	Inherited	Not applicable
Internal resistance	**Internal series resistance, R1**	Inherited	Not applicable
Battery charge capacity	Not applicable	`Finite`	Setting this parameter to `Infinite` disables other parameters.
Ampere-hour rating	**Ampere-hour rating**	Inherited	Not applicable
Charge	**Initial charge**	Inherited with priority `High`	This parameter for pre-R2017a models is an initializable variable on the **Variables** tab for 17a models.
Voltage V1 < Vnom when charge is AH1	**Voltage V1(< Vnom) when charge is Q1**	Inherited	Not applicable
Charge AH1 when no-load volts are V1	**Charge Q1 when no-load volts are V1**	Inherited	Not applicable
Model self-discharge resistance?	Not applicable	`Include`	Setting this parameter to

R2017a	Previous to R2017a	Value	Notes
			`Omit` disables the **Model self-discharge resistance?** parameter.
Self-discharge resistance	**Self-discharge resistance, R2**	Inherited	Not applicable
Model battery fade?	Not applicable	`Omit`	Setting this parameter to `Include` enables other parameters related to battery fading. For more information, see Battery.

Compatibility Considerations

The location of the Simscape™ language source file for the Battery block has changed.

If you are using the block in your models, there is no compatibility impact.

If you are using the source file as a member in your custom composite components (authored using Simscape language), update the member component declarations from `pe.sources.battery` to `elec.sources.battery`.

pe_getPowerLossSummary and pe_getPowerLossTimeSeries Functions: Calculate and view average or instantaneous power losses

Determine if devices represented by blocks are operating within power-loss requirements. Use the `pe_getPowerLossSummary` or the `pe_getPowerLossTimeSeries` function to calculate average or instantaneous power losses. The functions use logged Simscape data for *power_dissipated* variables.

To use either function, first enable simulation data logging and run the simulation. For more information, see "Data Logging" (Simscape). After simulation, call the function from the MATLAB® command line.

The `pe_getPowerLossSummary` function takes a Simscape logging node as the first input argument. The second and third input arguments are optional and represent the start and end of a time interval for averaging the power losses. If you omit these two input arguments, the function averages the power losses over the whole simulation time.

The `pe_getPowerLossSummary` function returns a MATLAB table. The first column lists all blocks within the specified logging node that have at least one *power_dissipated* variable, and the second column lists the corresponding losses in watts.

The `pe_getPowerLossTimeSeries` function also takes a Simscape logging node as the first input argument. The second and third input arguments are optional and represent the start and end of a time interval for averaging the power losses. If you omit these two input arguments, the function averages the power losses over the whole simulation time. You can specify a fourth argument. To have the function use a specific interval width for averaging the losses, specify the interval width as a fourth argument. To calculate the instantaneous power loss, specifying an interval width of zero.

The `pe_getPowerLossTimeSeries` function returns a MATLAB table and cell array. The first column of the table lists all nodes, within the specified logging node, that have at least one *power_dissipated* variable, and the second column lists the size of the time series data matrix in the cell array. The matrix contains three columns. The first two columns contain the interval start and end time. The third column contains the power-loss data in watts.

`pe_getEfficiency` Function: Calculate the efficiency of a circuit

Determine if your circuit is operating within efficiency requirements using the `pe_getEfficiency` function to calculate efficiency based on average power losses. The function uses logged Simscape data for *power_dissipated* variables.

To use the function, first enable simulation data logging and run the simulation. For more information, see "Data Logging" (Simscape). After simulation, call the function from the MATLAB command line.

The `pe_getEfficiency` function takes a sting that identifies the loads in your circuit as the first input argument. The Simscape logging node for the model is the second

input argument. The third and fourth input arguments are optional and represent the start and end of a time interval for averaging power losses. If you omit these two input arguments, the function averages the power losses over the whole simulation time.

The `pe_getEfficiency` function returns the efficiency of the circuit as a percentage based on the power dissipated by load and nonload blocks. If you specify a second output variable, the function also returns a MATLAB table. The first column lists all load blocks within the specified logging node that have at least one *power_dissipated* variable, and the second column lists the corresponding losses in watts.

power_dissipated Variable: Calculate and view instantaneous power dissipation

Blocks in the Semiconductor > Fundamental Components library and the Delta-Connected Load, Wye-Connected Load, and RLC blocks have an internal block variable called *power_dissipated*. This variable represents the instantaneous power dissipated by the block throughout the simulation. The data includes only the real power, not the reactive or apparent power, that the block dissipates. Semiconductor blocks that are composite components, for example the MOSFET block, have more than one *power_dissipated* variable.

To view and plot the instantaneous power dissipated by a component, first enable simulation data logging and run the simulation. Then, open the Simscape Results Explorer and navigate to the `power_dissipated` nodes for the variable of interest. For information on logging, viewing, and plotting simulation data using Simscape data logging and the Results Explorer, see "Data Logging" (Simscape).

Electric Drive Examples: Design algorithms and refine requirements for IPMSMs

These examples show interior permanent magnet synchronous machine (IPMSM) control strategies:

-
-
-

Voltage-Converter Control Examples: Control a push-pull buck converter using conduction mode control

These examples show you how to use discontinuous or continuous conduction mode (DCM or CCM) control to satisfy a voltage requirement using a push-pull buck converter:

-
-

Featured Examples

- The shows charging and discharging for a high-voltage battery. The model uses a realistic DC-link current profile.
- The example shows a cascaded control structure for a simple DC motor. A PWM-controlled four-quadrant chopper feeds the DC motor. The control system includes an outer loop for speed control and an inner loop for current control.

Specialized Technology

Automatic Simulation Type Selection: Inherit recommended solver settings from Simulink

In R2017a, when you select `auto (Automatic solver selection)` in your model, Simulink® manages the solver settings of the powergui block for optimal simulation performance. Selecting the `Variable-step` automatic solver sets the **Simulation type** to `Continuous` in the **Solver** tab of the powergui block. Selecting the `Fixed-step` automatic solver sets the **Simulation type** to `Discrete` in the **Solver** tab of the powergui block.

Selecting the `auto` solver mode of Simulink disables the solver settings of the powergui blocks. To tune the powergui solver settings based on the needs of your application, do not select the `auto` solver mode of Simulink.

Default Simulation Settings: Use recommended options for continuous and discrete simulation

When **Simulation type** is set to `Continuous`, the powergui block uses the ideal switching devices option by default. If you open an old model with **Use ideal switching devices** cleared, the classic `Continuous` mode is selected automatically. The **Use ideal switching devices** option is no longer available.

To simulate a model in the classic `Continuous` mode available in previous releases, select **Disable ideal switching** option in the **Preferences** tab. Do not select this option unless you experience simulation problems in the new `Continuous` mode. The **Preferences** tab now contains the continuous solver options that were available in the previous releases.

When **Simulation type** is set to `Discrete`, the powergui block uses the Tustin/Backward Euler (TBE) solver type by default. To simulate a model using the Tustin or Backward-Euler solver, specify the discrete solver in the **Preferences** tab. Unless you experience simulation problems in the `Discrete` mode or are developing models for real-time applications, keep the discrete solver set to Tustin/Backward Euler (TBE). The **Preferences** tab now contains the discrete solver options that were available in the previous releases.

Compatibility Considerations

If you open an old model with the **Use ideal switching devices** option cleared, the classic `Continuous` mode is selected automatically.

Battery Block Aging Model: Model degradation of lithium-ion battery performance over time

The Battery block now allows you to model the lifetime performance of a battery storage system for electric vehicles (EVs), hybrid electric vehicles (HEV), plugin HEVs, fuel cell vehicles (FCVs), and more electric aircraft (MOA). The new option provides a generic aging model with parameters that can be obtained from manufacturer datasheets or simple experiments. Simulate the impact of aging more easily and rapidly. The new option is available for the lithium-ion battery type only.

Electric Drive Blocks Modulation Parameterization: Configure drive to use hysteresis or space vector modulation (SVM)

The AC3, AC4, and AC6 blocks in the AC Drives library allow you to choose between hysteresis modulation or space vector modulation (SVM). To enable the integral proportional controller to use the difference between the target and measured current as input, on the **Controller** tab for the electric drive, for the **Modulation type** parameter, select `SVM`. Otherwise, select `Hysteresis`.

Electric Drive Block Output Labeling: Automatically apply signal names to bus measurements

On the **Asynchronous Machine** tab, electric drive blocks allow you to select **Use signal names as labels**. By default, when the option is not selected, the signals in the bus correspond to the descriptions of the measured signals and are organized into subsections. When the option is selected, the measurements in the output bus use generic variable names, listed in a single set with no subsections.

Featured Examples

This release includes two new examples and three converter topologies:

- The example shows the impact of aging, due to cycling, on a Lithium-Ion battery module.

- The example shows a Static Synchronous Compensator using 22 power modules per phase.
- The example shows the operation of a non-isolated Cuk converter.
- The example shows the operation of a flyback converter.
- The example shows the operation of a controlled buck converter.

R2016b

Version: 6.6

New Features

Bug Fixes

Simscape Components

Electrical Three-Phase Template: Shorten modeling process for three-phase power systems

The Electrical Three-Phase template is a complete Simscape Power Systems™ Simscape Components model example that you can use to:

- Create a three-phase model.
- Access Simscape and the Simscape single-phase and three-phase electrical libraries in the Simulink Library Browser.
- Explore simulation results using Simscape Results Explorer.
- Learn more about modeling using Simscape Power Systems Simscape Components.

To access the template, on the MATLAB Home tab, click **Simulink**. In the Simulink Start Page, click **Simscape**, and then click the Electrical Three-Phase template. A new model using the template settings and contents opens in the Simulink Editor.

Thermal Variants for Semiconductor Blocks: Model heat loss due to conduction losses and switching events

Thermal ports allow you to model the heat that switching events and conduction losses generate in these semiconductor blocks:

- Commutation Diode
- Diode
- GTO
- MOSFET
- Thyristor

The thermal ports are optional and are hidden by default. To expose a thermal port, right-click the block in your model and, from the context menu, select **Simscape** > **Block choices**. Choose a variant that includes a **Thermal port**. This action displays the thermal port on the block and allows you to specify thermal parameters based on manufacturer-supplied data.

The semiconductor blocks model thermal losses using your choice of two different parameterization methods:

- **Voltage and current**
- **Voltage, current, and temperature**

The IGBT semiconductor block now also offers you the option to model thermal losses using either parameterization method.

Supercapacitor Source: Model high power density source that has rapid charging and discharging capability

The Supercapacitor block models a voltage-dependent, electrochemical double-layer capacitor (ELDC). You can use a single Supercapacitor block to represent any number of supercapacitor cells connected in series or in parallel.

For an example that uses the Supercapacitor block, see Supercapacitor Charging and Discharging Behavior.

Control Library with Three-Phase PWM Blocks: Control power converters using three-phase, two- and three-level pulse width modulation

The new Control library includes a Pulse Width Modulation (PWM) sublibrary that contains the PWM Generator (Three-phase, Three-level) and PWM Generator (Three-phase, Two-level) blocks. You can use the PWM generator blocks to generate gate pulses for power conversion control during desktop simulation or to deploy modulation waveforms for simulating on target hardware. Both blocks perform continuous PWM generation using the modulation algorithm and sampling mode that you specify. The PWM Generator (Three-phase, Two-level) block can also perform discontinuous PWM generation.

For examples that use the PWM generator blocks, see:

- Three-Phase Three-Level PWM Generator
- Three-Phase Two-Level PWM Generator

Both examples also use the Spectrum Analyzer block from the Simscape Utilities library. The Spectrum Analyzer block accepts input signals with discrete sample times and displays frequency spectra of these signals. It allows you to perform measurements and analyze the signals using a variety of methods.

Featured examples

This version includes two additional new examples that show methods for controlling three-phase asynchronous machines. The new examples, which also show how to use the Spectrum Analyzer block from the Simscape Utilities library, are:

- Three-Phase Asynchronous Drive with Sensor Control
- Three-Phase Asynchronous Drive with Sensorless Control

A new version of the Harmonic Analysis of a Three-Phase Rectifier example also uses the Simscape Spectrum Analyzer block.

Specialized Technology

Multilevel PWM Block: Control power conversion using multilevel pulse width modulation

Use the PWM Generator (Multilevel) block in the Control & Measurements library to generate pulses for PWM-controlled modular multilevel converters. Use the generator block to control switching behavior in half- or full-bridge modular, multilevel power converter blocks.

Modular, Multilevel Converter Bridge Blocks: Convert power using half and full MMC bridge blocks

Use the Half-Bridge MMC and Full-Bridge MMC blocks from the Power Electronics sublibrary of the Fundamental Blocks library to simulate power conversion in high-voltage transmission systems. Each block consists of multiple series-connected power modules. You can control the blocks using pulse width modulation (PWM), PWM-dependent switching functions, or reference-signal-dependent switching functions. The Half-Bridge MMC block also allows you to control power conversion using an aggregate model.

Featured examples

New examples in this version are:

- The Home Energy Management System example shows how to model a small-scale, home-based power system.
- The Switching Function Converter Controlled by Averaged Firing Pulses example shows the advantage of modeling power conversion using switching functions and averaged pulses.

R2016a

Version: 6.5

New Features

Bug Fixes

SimPowerSystems renamed to Simscape Power Systems

The product name SimPowerSystems™ is changed to Simscape Power Systems.

The top-level block library for the product is still located under **Simscape** in the Library Browser, but is now named **Power Systems**. To open the top level-block library using the MATLAB command prompt, enter `simscapepowersystems`.

Simscape Components

Thermal Port Option for IGBT Block: Model heat losses in insulated-gate bipolar transistors

Thermal ports allow you to model the heat that switching events and conduction losses generate in an IGBT block. The thermal port is optional and is hidden by default. To expose the thermal port, right-click the IGBT block in your model and, from the context menu, select **Simscape** > **Block choices**. Choose a variant that includes a **Thermal port**. This action displays the thermal port on the block and adds a **Thermal Model** tab to the block dialog box.

Thermal Library: Model thermal behavior using equivalent thermal circuit and thermal resistance blocks

The Thermal library contains three blocks for modeling heat transfer:

- Cauer Thermal Model Element — An equivalent thermal circuit that, in a series connection, models heat transfer as a function of the thermal characteristics of the individual physical components and materials, for example, a chip, solder, and base, that comprise a semiconductor.

- Foster Thermal Model — An equivalent thermal circuit that models heat transfer as a function of the thermal characteristics of a semiconductor.

- Thermal Resistor — A thermal interface resistance circuit that models conductive heat transfer through a layer of material. Use the Thermal Resistor block to parameterize heat transfer using the thermal resistance value of the material.

Mutual Resistance Parameter for Transmission Line Block: Model line-to-line resistance in long cable runs

To model mutual resistance, open the block dialog box of the Transmission Line block and on the **Main** parameter tab, for the **Mutual resistance** parameter, specify the line-to-line resistance per unit length.

Featured example

The Quantifying IGBT Thermal Losses example shows how to generate a loss-dependent temperature profile using an IGBT block thermal variant with a Cauer or Foster thermal model.

Specialized Technology

Battery Block Heat Loss Model: Specify separate heat loss data for charging and discharging batteries

The **Heat loss difference [charge vs discharge] (W)** parameter allows you to specify the difference in thermal energy dissipation for charging versus discharging for the Lithium-ion variant of the Battery block.

Battery Cut-Off Voltage Parameter: Specify minimum allowable battery voltage

The **Cut-off Voltage (V)** parameter allows you to specify the minimum allowable battery voltage, which represents the end of the discharge characteristic for the Battery block.

Specialized Technology main library renamed to simscapepowersystems_ST

The main Specialized Technology library is renamed from simpowersystems_ST to simscapepowersystems_ST. To open the simscapepowersystems_ST library using the MATLAB command prompt, enter `simscapepowersystems_ST`.

R2015b

Version: 6.4

New Features

Bug Fixes

Simscape Components

Single-Phase Circuit Breaker (with arc) block

The Single-Phase Circuit Breaker (with arc) block represents an AC circuit breaker with Mayr arc representation. The breaker is closed when the input control voltage, vT, is below the threshold value that you specify. If the control voltage rises above the threshold, the circuit opens with an arc during the current interruption. The breaker can open and close repeatedly.

Nonlinear Transformer and Nonlinear Inductor blocks

Two new blocks in the Passive Devices library let you consider nonlinearities in inductors and transformers due to magnetic saturation:

- The Nonlinear Inductor block represents an inductor with a core that is nonideal, due to its magnetic properties or dimensions. There are five options for block parameterization:

  - `Single inductance (linear)`
  - `Single saturation point`
  - `Magnetic flux versus current characteristic`
  - `Magnetic flux density versus field strength characteristic`
  - `Magnetic flux density versus field strength characteristic with hysteresis`

  The last option is based on the Jiles-Atherton model of hysteresis.
- The Nonlinear Transformer block is based on the Nonlinear Inductor block and has similar parameterization options, which let you model varying levels of nonlinearity. You can parameterize the transformer winding either by combined or by separate primary and secondary values for leakage resistance and inductance.

Back EMF profile parameterization for Brushless DC Motor block

The Brushless DC Motor block has four options for defining the permanent magnet flux distribution as a function of rotor angle:

- `Perfect trapezoid - specify maximum flux linkage`

- `Perfect trapezoid - specify maximum rotor-induced back emf`
- `Tabulated - specify flux partial derivative with respect to rotor angle`
- `Tabulated - specify rotor-induced back emf as a function of rotor angle`

The first two options are for simple parameterization. You specify maximum values and the block assumes a perfect trapezoid for the back emf. The second two options allow you to parameterize by specifying the flux linkage partial derivative or the measured back emf constant for a given rotor speed. Use the tabulated options for more accurate results.

Featured examples

New examples in this version are:

- The Three-Phase Asynchronous Direct Online Motor Connected to Hydraulic Pump example shows how valve diameter affects the speed and current that the motor draws.
- The Three-Phase Synchronous Machine Governor Control Design example shows how to linearize a SimPowerSystems Simscape Components model for control system stability, analysis, and design.
- The Electric Power Assisted Steering example shows how to use a permanent magnet synchronous motor to amplify driver-applied force in a power-assisted automobile steering system.

Specialized Technology

Fundamental Drive Blocks library

The Electric Drives library now contains a new sublibrary, named Fundamental Drive Blocks, with 22 new blocks. The new blocks include current and speed controllers, bridge firing units, rectifiers, modulators, and generators. These blocks represent elements that are used to build the AC and DC electric drive blocks of the Electric Drives library. You can use the fundamental drive blocks to build your own electric drive models.

Power converter blocks

The Power Electronics library now contains seven new blocks that represent common power converter topologies. The blocks can simulate three types of modeling-level details:

- The switching devices mode, in which the converters are modeled using ideal power switches and antiparallel and clamping diodes
- The switching function mode, in which voltage sources and current sources represent the switches
- The average mode, in which the converters use reference signals as control signals to generate the desired voltages at the output terminals of the converter

Each type addresses a particular need for accurate and fast simulation performance.

The example shows several types of power electronics converters. You can simulate these converters using any of the four selectable modeling techniques.

Load flow analysis for systems with unbalanced currents or single-phase connections

The Load Flow tool now allows a balanced or unbalanced load flow analysis on three-phase, two-phase, single-phase, or a mix of 3-2-1-phase circuits. An improved version of the Load Flow Bus block is the key element used to identify and set up a one-phase, two-phase, or three-phase load flow bus. The current three-phase load flow blocks (Machines, RLC Loads, etc.) can now specify unbalanced load flow settings. Additional load flow blocks (single-phase RLC Loads and AC Voltage Source blocks) now have load flow setup variables on the **Load Flow** tab.

The IEEE 13 Node Test Feeder example shows how to use unbalanced load flow analysis to initialize the IEEE 13 Node Test Feeder circuit.

External temperature input for Battery block

The Battery block has been improved to include the impact of ambient temperature. The temperature-dependent battery model represents the performance of common battery types. An option in the block allows you to enable or disable the modeling of temperature effect. The **Temperature** tab in the mask (for the Lithium-Ion battery type only) groups all the parameters related to the temperature-effect model.

The Lithium-Ion Temperature Dependent Battery Model example shows the impact of temperature on a 7.2 V, 5.4 Ah, Lithium-Ion battery module.

Flickermeter block and statistical module function

The new Digital Flickermeter block, found in the Control and Measurements/ Measurement library, implements a digital flickermeter as described in the IEC 61000-4-15 standard. The model implements the first four modules of the design. A new function for offline analysis implements the fifth module. The new function is power_flicker. You can use it at the command prompt or access it from the mask of the Digital Flickermeter block.

The Flickermeter Statistical Analysis Module example shows a method for computing short-term flicker severity of a phase-to-ground voltage.

Powergui enhancements

The powergui block has a new option that is associated with the interpolation option introduced in R2014b. The option allows you to provide time-stamped gate signals to power electronic devices, in a hardware-in-the-loop context.

The **Allow multiple Powergui blocks** option is no longer required. The software detects any incompatible use of powergui blocks in a model.

Featured examples

New examples in this version include:

- Power Flow Control and Line Deicing Using a Bundle-Controlled Line Impedance Modulator (LIM)
- Gear Speed Reducer (Simscape model)
- Shaft Coupling (Simscape model)

R2015a

Version: 6.3

New Features

Bug Fixes

Simscape Components

Asynchronous machines with SI parameterization

The Asynchronous Machine Squirrel Cage (fundamental, SI) and the Asynchronous Machine Wound Rotor (fundamental, SI) blocks allow you to use SI values for parameterization.

Synchronous Machine Model 2.1 blocks

The Synchronous Machine Model 2.1 (fundamental) and the Synchronous Machine Model 2.1 (standard) blocks use a simplified version of Park's transform and a simplified representation suitable for large-scale stability studies. The blocks neglect the effect of speed variation on stator voltages.

Zigzag-Delta1-Wye and Zigzag-Delta11-Wye Transformer blocks

The Zigzag-Delta1-Wye Transformer and Zigzag-Delta11-Wye Transformer blocks model linear, nonideal transformers that you can use to reduce the total harmonic distortion in your model. The blocks have three-limb cores with primary windings configured in a zigzag connection. The secondary windings have delta (1 or 11 o'clock) and wye configurations.

Average-Value Inverter block

The Average-Value Inverter block models an average-value, full-wave inverter that converts DC voltage to three-phase AC voltages and converts three-phase AC power demand to DC power demand. The Average-Value Inverter block does not yield the harmonics that are typically associated with a detailed inverter representation. The block converts power with fixed losses.

Ideal Rectifier block name change

The Ideal Rectifier block name has changed to Average-Value Rectifier.

Featured examples

Four examples are introduced in this version:

- The Custom Inductor (B-H Curve) example compares the behaviors of linear and nonlinear inductors. The inductors are implemented in the Simscape language.
- The Custom Transformer (B-H Curve) example shows the calculation and confirmation of a nonlinear transformer core magnetization characteristic. The transformers are implemented in the Simscape language.
- The Marine Full Electric Propulsion Power System example shows a representative marine electrical power system with base load, hotel load, bow thrusters and an average-value propulsion rectifier.
- The Three-Phase Custom Transforms example compares the results of four different three-phase transforms.

Specialized Technology

PV Array block

The PV Array block allows you to model series-connected and parallel-connected PV modules. You can use a preset PV module from NREL System Advisor Model or your own PV modules.

These examples use the PV Array block:

- 250-kW Grid-Connected PV Array
- 400-kW Grid-Connected PV Farm (Average Model)
- Partial Shading of a PV Module
- Single-Phase, 240 Vrms, 3500 W Transformerless Grid-Connected PV Array

Annotation and export options for Load Flow Tool

The Load Flow Tool has a new option to annotate the load flow calculations for models that contain a Load Flow Bus block. To use this option, at the MATLAB command prompt, enter `power_loadflow('-v2',gcs,'AddBuses')`. You can also add the annotation using the powergui block in your model. To open the tool, open the powergui block. In the dialog box, on the **Tools** tab, click **Load Flow**, and then click **Add bus blocks**.

The Load Flow Tool also has a new option to generate reports. To open the tool, open the powergui block. In the dialog box, on the **Tools** tab, click **Load Flow**, and then click **Compute**. When the dialog box displays `Load Flow Converged!`, click **Report**. Specify the report format and save the `LoadFlowResults_power_machines` file. When the load flow report dialog box indicates that your load flow report has been created, click **Open file** to view the report.

`power_customize` function for creating custom Specialized Technology blocks

The power_customize function helps you create custom SimPowerSystems Specialized Technology blocks. To open the function, open the powergui block. In the dialog box, on the **Tools** tab, click **Customize SPS blocks**. You can also open the tool by entering `power_customize` at the MATLAB command prompt.

Three-limb core option for three-phase transformer blocks

In prior releases, the Three-Phase Transformer (Three Windings) and Three-Phase Transformer (Two Windings) blocks modeled three-phase transformers using three separate single-phase transformers. Now you can model a transformer using a single three-limb or five-limb core representation for high-validity results. Use the **Core type** parameter on the **Configuration** tab of either three-phase transformer block dialog box to specify the representation.

These examples show the validity of core representation simulation results:

- Three-Phase Three-Limb (Core-Type) Two-Winding Transformer
- Three-Phase Five-Limb (Shell-Type) Three-Winding Transformer

Interpolation and Store State-Space Matrices options for Tustin solver

The discrete Tustin solver on the powergui block has two options for increasing simulation speed without sacrificing simulation accuracy. The interpolation option increases simulation speed by enabling the solver to interpolate to identify power electronic device gate transitions that occur between large sample times. To use the interpolation option, open the powergui block. In the dialog box, on the **Solver** tab, select Discrete for the simulation type and Tustin for the solver type. Select the **Interpolate** check box.

The **Store state-space matrices** option accelerates simulation by enabling the solver to store and reuse matrix computation results. The option is visible only when the simulation type is set to Discrete, the solver type is set to Tustin, and the **Interpolate** option is selected. Select **Store state-space matrices** and specify a buffer size.

New machine block dialog boxes and function access

The **Configuration** tab on the Asynchronous Machine block dialog box now contains a **Preset parameters** section. The section includes the **Squirrel-cage preset model** parameter (formerly called **Preset model**). It also includes the **Double squirrel-cage preset model** parameter, which has an option to open the parameter estimator (the power_AsynchronousMachineParams function) for estimating machine parameters based on manufacturer specifications. The **Rotor type** parameter is now the first parameter on the **Configuration** tab and is no longer disabled when a preset model is selected.

The Permanent Magnet Synchronous Machine block dialog box **Advanced** tab now contains a **Machine parameters** section. It includes a **Compute from standard manufacturer specifications** option. The option opens the `power_PMSynchronousMachineParams` function.

New powergui dialog box and tools

You can use the new **Tools** tab on the powergui block to open these tools:

· Steady-State Voltages and Currents Tool

· Initial States Tool

· Load Flow Tool

· Machine Initialization Tool

· Impedance vs Frequency Measurement Tool

· FFT Analysis Tool

· Linear System Analyzer

· Hysteresis Design Tool

· Compute RLC Line Parameters Tool

· Generate Report Tool

· power_customize

The parameters that were on the **Load Flow** tab are now included on the **Preferences** tab on the powergui block. The **Preferences** dialog box also includes a new option that allows you to use a different powergui block in each subsystem in your model.

Block library changes

The powerlib library is no longer the main SimPowerSystems Specialized Technology library. The main library is now the simpowersystems_ST library. To access the simpowersystems_ST library through the Simulink Library Browser, go to **Simscape** > **SimPowerSystems** > **Specialized Technology**. To open the main library at the MATLAB command prompt, enter `simpowersystems_ST`.

The simpowersystems_ST library contains these libraries:

· Fundamental Blocks

· Control & Measurements

- Electric Drives — Formerly included in the Applications library.
- FACTS — Formerly named the Flexible AC Transmission Systems (FACTS) library and included in the Applications library.
- Renewables — Formerly named the Renewable Energy library and included in the Applications library.

You can access the Fundamental Blocks library in the Simulink Library Browser, or by entering, `powerlib` at the command prompt. The Fundamental Blocks library contains the powergui block. It also contains these libraries that were previously included in the powerlib library:

- Electrical Sources
- Elements
- Interface Elements
- Machines
- Measurements
- Power Electronics

The Solar library is a new library that contains the PV Array block. The Solar library is included in the FACTS library.

The Excitation Systems library is included in the Machines library.

The Applications library is no longer included in the Specialized Technology library.

Phasor Simulation Method examples

The Simplified Model of a Small Scale Micro-Grid and 24-hour Simulation of a Vehicle-to-Grid (V2G) System examples use the phasor simulation method to accelerate micro-grid simulations.

R2014b

Version: 6.2

New Features

Bug Fixes

Simscape Components

Harmonics option in Voltage Source block

The Voltage Source block has a **Harmonics** tab that allows you to configure the block as a three-phase voltage source that maintains a fundamental frequency and harmonics across its terminals. You can specify the harmonic orders and the harmonic to peak magnitude ratios of the AC voltage waveform.

Wye-Connected Variable Load blocks

The Wye-Connected Variable Load and Wye-Connected Variable Load (lagging) blocks model three-phase, wye-wired configurations that are capable of representing real, or real and reactive, time-varying loads. You parameterize the blocks via their physical input signals.

The Wye-Connected Variable Load block calculates the resistance required to dissipate the real power that its physical input signal specifies. The Wye-Connected Variable Load (lagging) block calculates the resistance and inductance required to dissipate the real and reactive powers that its physical input signals specify.

Three-Level Converter block

The Three-Level Converter block models a twelve-pulse, three-phase, three-level controlled converter circuit that you can use to connect a three-phase AC network to a three-level DC network. Each of its three bridge arms contains four switching devices and the associated anti-parallel diodes. You can use any of these blocks for the switching devices:

- GTO
- Ideal Semiconductor Switch
- IGBT
- MOSFET

You use the Twelve-Pulse Gate Multiplexer block to multiplex the connections into a single vector to provide input to the Three Level Converter block.

Machine Mechanical Power and Synchronous Machine Field Circuit interface blocks

The machine mechanical power blocks allow you to specify mechanical power to or from a machine block using either an SI, Machine Mechanical Power (SI), or per-unit, Machine Mechanical Power (pu), physical signal. The machine power blocks contain mechanical rotational reference and machine inertia components in order to accurately represent transient behavior.

The synchronous machine field circuit blocks allow you to use a physical signal to apply either SI, Synchronous Machine Field Circuit (SI), or per-unit, Synchronous Machine Field Circuit (pu), voltage to a machine field circuit. It also allows you to use a physical signal to measure the current in the field circuit.

Ideal Rectifier block

The Ideal Rectifier block models an ideal, full-wave, six-pulse rectifier that converts three-phase AC voltages to DC voltage and DC power demand to three-phase AC power demand. The Ideal Rectifier block does not yield the harmonics that are typically associated with a detailed rectifier representation. It converts power ideally to provide a faster simulation than the non-idealized Rectifier block provides.

Removal of Initial state parameter on Circuit Breaker block

In previous releases, there was an **Initial state** parameter on the dialog box of the Circuit Breaker and Single-Phase Circuit Breaker blocks. You set the parameter to `open` or `closed` to indicate the state of the breaker at the start of the simulation. The breaker blocks now calculate the initial state as a function of the input control voltage, `vT`, at `time = 0`. Therefore, you are no longer required to set the initial state parameter.

New examples

Examples introduced in this version are:

- Three-Phase Asynchronous Wind Turbine Generator
- Three-Phase Custom Simplified Synchronous Machine

Specialized Technology

Tustin/Backward Euler discretization method

The powergui block provides a discretization solver option that combines the advantages of the Tustin and the Backward-Euler methods in a single method. The Tustin/Backward Euler (TBE) solver is now the recommended discretization method for all discrete simulation.

Bipolar PWM option in PWM Generator (2-Level) block

You can specify the generator type for the PWM Generator (2-Level) block to use either unipolar or bipolar PWM modulation for single-phase full-bridge control. The single-phase full-bridge generator uses unipolar modulation, which produces higher quality AC waveforms. The single-phase full-bridge, bipolar generator uses bipolar modulation, which produces low-variation, common-mode voltage.

The block allows you to choose from three different techniques to sample the reference signal Uref. Natural sampling models the behavior of an analog implementation of a PWM generator. Asymmetrical regular sampling is a single-edge technique that samples Uref only at the valley of the carrier. Symmetrical regular sampling is a double-edge technique that samples Uref at both the valley and the peak of the carrier.

Parameter estimation for Stern model in Supercapacitor block

The Supercapacitor block no longer requires you to specify hard-to-find values for surge voltage, leakage current on the **Parameters** tab. The block uses default values for these parameters and for the overpotential and charge transfer parameters on the **Stern** tab. You can set the **Stern** parameter to values that you find, via trial-and-error simulation (or data sheet specification testing using the Optimization Toolbox™), to match the performance of your supercapacitor.

The block can simulate your supercapacitor's self-discharge characteristic using current and voltage information that you typically find on a data sheet.

Usability enhancements for Switched Reluctance block

The Switched Reluctance Motor block offers you three preset motors types. The block contains all of the model parameters that it needs to simulate a 10, 60, or 75 kW motor.

You can specify a model of your motor by entering the specifications directly in the dialog box or by referring to them in a MATLAB MAT-file.

Initialization enhancements for examples using initial condition vectors

These renewable energy examples use an initial state vector to start in steady state:

- Wind Farm (IG)
- Wind-Turbine Asynchronous Generator in Isolated Network
- Wind Farm (DFIG Phasor Model)
- Wind Farm - DFIG Detailed Model
- Wind Farm - DFIG Average Model
- Wind Farm - Synchronous Generator and Full Scale Converter (Type 4) Detailed Model
- Wind Farm - Synchronous Generator and Full Scale Converter (Type 4) Average Model

When you make changes to the model, you need to regenerate or disable the initial state vector to avoid getting a Simulink error when you start the simulation. Follow the instructions in the initialization file in the example's model properties to regenerate the initial state vector.

R2014a

Version: 6.1

New Features

Bug Fixes

Compatibility Considerations

New names for SimPowerSystems technologies

SimPowerSystems Version 6.1 introduces new names for the SimPowerSystems Second and Third Generation technologies. SimPowerSystems Simscape Components is the new name for the Third Generation technology. SimPowerSystems Specialized Technology is the new name for the Second Generation technology.

Note: The terms Simscape Components and Specialized Technology refer to the SimPowerSystems technology used. A SimPowerSystems *version* can contain one or both technologies.

For more information, see Comparison of Simscape Components and Specialized Technology.

Simscape Components

Time-Based Fault and Enabled Fault blocks

The Time-Based Fault block and the Enabled Fault block model any permutation of a single-phase, two-phase, or three-phase grounded or ungrounded fault. You specify the fault permutation and the values for resistance and conductance.

The Time-Based Fault block is activated and deactivated during simulation according to time-based parameters that you specify. The Enabled Fault block is activated when the input signal exceeds a threshold value that you specify. It is deactivated when the signal is equal to or less than the threshold value.

Saturation option for Synchronous Machine models

You can now simulate magnetic saturation on these synchronous machine blocks:

- Synchronous Machine Round Rotor (fundamental)
- Synchronous Machine Round Rotor (standard)
- Synchronous Machine Salient Pole (fundamental)
- Synchronous Machine Salient Pole (standard)

These blocks have a new **Saturation** tab, which contains the **Magnetic saturation representation** parameter. When you specify the field current and air-gap voltage per-unit saturation data, the block generates a per-unit air-gap voltage versus field current open-circuit lookup table. The block then uses the lookup table data to calculate a saturation factor.

Harmonic analysis functions

There are three new functions for performing harmonic analysis:

- pe_getHarmonics
- pe_calculateThdPercent
- pe_plotHarmonics

The `pe_getHarmonics` function returns the harmonic orders, magnitude, and fundamental frequency when you input a Simscape logging node variable.

To run the `pe_getHarmonics` function, at the MATLAB command prompt, type:

```
pe_getHarmonics( logging_node )
```

where *logging_node* is the Simscape logging node variable.

The `pe_calculateThdPercent` function returns the Total Harmonic Distortion (THD) percentage when you use the harmonic order and the harmonic magnitude as input arguments.

To run the `pe_calculateThdPercent` function, at the MATLAB command prompt, type:

```
pe_calculateThdPercent( harmonicOrder, harmonicMagnitude )
```

where

- *harmonicOrder* is a vector of harmonic orders.
- *harmonicMagnitude* is a vector of harmonic magnitudes.

The `pe_plotHarmonics` function plots the harmonic data when you input a Simscape logging node variable.

To run the `pe_plotHarmonics` function, at the MATLAB command prompt, type:

```
pe_plotHarmonics( logging_node )
```

where *logging_node* is the Simscape logging node variable.

The Harmonic Analysis of a Three-Phase Rectifier example shows you how to use the three functions.

Primary and secondary winding blocks with new variables tab

The Primary Winding and Secondary Winding blocks now contain a **Variables** tab, which allows you to specify target value and priority for a new initialization process.

For more information see the Simscape Release Notes item, Variables tab for specifying target value and priority for new initialization process

Compatibility Considerations

In previous releases, the Primary Winding and Secondary Winding blocks contained parameters that let you specify an initial value for some internal block variables at the

start of simulation. These parameters have now been removed. The following table lists the initialization parameters that have been removed from block dialogs and the names of the corresponding block variables:

Block Name	Parameter Name	Variable Name
Primary Winding	**Leakage inductance initial current** **Magnetization inductance initial current** **Initial magnetic flux**	**Leakage inductance current** **Magnetization inductance current** **Magnetic flux**
Secondary Winding	**Leakage inductance initial current** **Initial magnetic flux**	**Leakage inductance current** **Magnetic flux**

Legacy models using these blocks are affected by this change. If a block used the initialization parameter, then, once you open the model in the current release, this parameter value is no longer visible and is not automatically mapped to the corresponding variable value. The simulation results will only stay the same if you set the target value of the variable to be that of the original parameter and set the parameter priority to High.

Specialized Technology

Aircraft fuel cell hybrid emergency power system example

The Energy Management Systems for a Hybrid Electric Source (Application for a More Electric Aircraft) example illustrates a simulation model of a fuel-cell-based emergency power system for More Electric Aircraft (MEA). In MEA, as the landing-gear and flight-control systems become more electrically driven, the peak electrical load seen by the emergency power system increases. This increased load puts conventional ram air turbine (RAT) and air-driven generator (ADG) emergency power systems, which exhibit near-zero power production at lower speeds, at risk for overloading. This example presents an alternative emergency power system based on fuel cells, lithium-ion batteries, and supercapacitors that is more capable of handling the increased electrical load.

Text labels for Machine Block Bus output signals

In previous releases, you could only use alphanumeric signal names to identify bus labels on these SimPowerSystems machine blocks:

- Asynchronous Machine SI Units
- Asynchronous Machine pu Units
- DC Machine
- Permanent Magnet Synchronous Machine
- Simplified Synchronous Machine SI Units
- Simplified Synchronous Machine pu Units
- Single Phase Asynchronous Machine
- Synchronous Machine SI Fundamental
- Synchronous Machine pu Fundamental
- Synchronous Machine pu Standard

Machine blocks now have a **Measurement output** parameter that gives you the option to identify bus labels with alphanumeric signal names that are compatible with model referencing. To enable the option to convert bus signal names to valid signal names, on the machine block **Configuration** tab, under **Measurement output**, select the **Use signal names to identify bus labels** check box.

The **Use signal names to identify bus labels** check box is cleared by default.

Three-phase 3-Level Inverter with direct specification of diode and IGBT characteristics

In previous versions, the Loss Calculation in a Three-Phase 3-Level Inverter example allowed you to choose between three different pre-specified commercial (ABB® or Fuji) components for the **IGBT Type** and **Diode Type** parameters in the Half-bridge IGBT with Loss Calculation block dialog box. To use a value other than the prespecified commercial values, you entered your own IGBT and diode specifications via MATLAB MAT-files.

In the current version, you can enter your preferred IGBT and diode values directly to the Half-bridge IGBT with Loss Calculation block dialog box in the new **IGBT** and **Diode** tabs. You can click the **Save** button on either tab to save the values to MAT-files for future use. The tabs also contain a **Plot Characteristics** button that you can use to plot your IGBT or diode values.

The prespecified commercial (ABB or Fuji) component specifications are still available as MAT-files that you can upload by clicking the **Load** button on the **IGBT** or **Diode** tab.

Improved breaker block set interfaces

The dialog boxes for the breaker blocks in the Elements library have been standardized:

- Breaker
- Three-Phase Breaker
- Three-Phase Fault

`power_cableparam` function documentation enhancement

The power_cableparam function reference page has been expanded to include a formula for calculating the *Total Section* of a screen conductor.

R2013b

Version: 6.0

New Features

Bug Fixes

Third Generation

Third Generation block libraries based on Simscape technology

SimPowerSystems Version 6.0 introduces Third Generation technology to model, simulate, and analyze electrical power systems. Third Generation technology provides libraries of component models fully compatible with the Simscape Foundation library. These component models are also fully compatible with Simscape technology, including local solver and data logging.

Version 6.0 provides full support for both Second Generation and Third Generation functionality, including block sets and tools. With Version 6.0, you have the option to create and simulate a model using Second or Third Generation technology. Version 6.0 software gives you the two full block libraries, labeled Second Generation (SPS 2G) and Third Generation (SPS 3G). The Second Generation (SPS 2G) libraries contain all the blocks available prior to Version 6.0, and the Third Generation (SPS 3G) libraries contain 55 new blocks.

> **Note:** The term *generation* — e.g., "Third *Generation*" — refers to the SimPowerSystems technology used. A SimPowerSystems *version* can contain one or both *generations*.

For more information, see Comparison of Second and Third Generation Technologies.

Expandable three-phase electrical ports for single-line diagrams

Third Generation library blocks are equipped with the new three-phase conserving electrical ports, which you can individually expand and collapse. Three-phase ports on Third Generation blocks are collapsed by default, to support single-line diagrams. You can optionally expand them to separate the phases, for example, if you need to inject a single-line-to-ground fault into your circuit. For more information, see Three-Phase Ports.

Machines and transformers customizable using Simscape language, with examples

The ThreePhaseExamples library, included in the product examples, contains the following custom three-phase components:

- Permanent Magnet Synchronous Motor
- Synchronous Machine
- Zigzag Transformer

You can use these simplified example models to write your own machine and transformer component files.

To open the custom library, type `ThreePhaseExamples_lib` at the MATLAB Command prompt. Double-click any block in the library to open its dialog box, and then click the `View source for` *BlockName* link in the block dialog box to open the Simscape source file for this block in the MATLAB Editor.

To customize the block for your application, edit the source file and save it in a package directory.

For more information on writing customized component files, see Custom Components. For information on packaging and deploying Simscape component files, see Simscape File Deployment.

Second Generation

Electric Drive blocks with improved measurement outputs

A new parameter, **Output bus mode**, has been added to all the AC and DC drives in the Electric Drives library. The parameter lets you select between two options:

- `Multiple output buses` — The block has separate `Motor`, `Conv`, and `Ctrl` outputs, as before. This is the default.
- `Single output bus` — The three separate outputs are combined into a single bus output called `Out`.

Examples that use blocks from the Control and Measurements library

Example models that were using blocks from the Extras library have been modified to use the blocks from the Control and Measurements library, introduced in R2013a.

New examples

Examples introduced in this version are:

- 100-kW Grid-Connected PV Array (Detailed Model)
- 100-kW Grid-Connected PV Array (Average Model)

R2013a

Version: 5.8

New Features

Bug Fixes

Compatibility Considerations

Control and measurement blocks automatically configured based on simulation mode

A new Control and Measurements library, with 47 new blocks, provides improved control and measurement blocks for use in power electrical models. The new library is a restructured and enhanced version of the old Extras library. All the blocks in the new library have associated block reference pages that refer to small simulation examples. New design techniques are used to improve model robustness and efficiency. Compared to the old Extras library, the new Control and Measurements library eliminates duplicate of continuous, discrete, and phasor versions of the same block, by merging them into a single block. These blocks feature a simple mechanism that automatically configures the block based on simulation mode.

Compatibility Considerations

The Control and Measurements library is intended to replace the Extras library. The Extras block library is no longer listed in the Library Browser. It is still available for compatibility purposes, and you can open it by typing `powerlib_extras` at the MATLAB command prompt.

If your legacy models contain blocks from the Extras library, they will continue to work. However, MathWorks® recommends that you use the blocks from the Control and Measurements library in your new models.

Supercapacitor block

The Supercapacitor block in the Electric Drives library implements a generic model parameterized to represent most popular types of supercapacitor. The Supercapacitor Model example illustrates a simple hybridization of a supercapacitor with a battery.

power_analyze function that now calculates state-space matrix based on switch state

The power_analyze function now has an option that allows you to compute ABCD matrices of a given model based of any given switch status vector provided as an input argument.

Additional diagnostic controls for SimPowerSystems models

The **Preferences** tab of the powergui block parameters dialog box has been enhanced to give you better control on what you want to see displayed in the command window during simulation. The **Display SimPowerSystems warnings and messages** check box, available in previous releases, is now replaced with two new check boxes:

- **Disable SimPowerSystems warnings** allows you to deactivate all the warning messages that are displayed when you run a simulation. The warning messages are displayed by default. Select this check box if you are aware of the warning and do not want to be bothered by a message on subsequent simulation runs.
- **Display SimPowerSystems compilation messages** allows you to display the compilation messages issued during the analysis of the model. These messages are not displayed by default.

New examples

Examples introduced in this version are:

- Loss Calculation in a Three-Phase 3-Level Inverter
- Supercapacitor Model

R2012b

Version: 5.7

New Features

Bug Fixes

Seven IEEE automatic voltage regulator (AVR) excitation blocks

The Machines library now contains a sublibrary, named Excitation Systems, with seven new blocks. These blocks represent the standard types of IEEE® AVR excitation systems. For more information, see the block reference pages.

The `power_machines` example model, which previously used a generic excitation system, has been updated to use any of the seven models in the Excitation Systems (`AVRlib`) library.

`power_new` function that creates new Simulink model with recommended settings for SimPowerSystems models

When you type `power_new` at the MATLAB Command prompt, the software creates a new model, with the recommended solver `ode23tb` and with required Solver Configuration and powergui blocks already on the canvas. It also opens the `power_new_palette` library.

After using `power_new`, continue developing your model by copying the blocks from the `power_new_palette` library, as needed, and adding other blocks from the Simulink and Simscape libraries.

Optional Simscape mechanical rotational port for SimPowerSystems electric drives models

Blocks in the Electric Drives library now have an additional option, `Mechanical rotational port`, for the **Mechanical input** parameter. When you select this option, the mechanical input port of the block (`Tm` or `w`), changes to a Simscape conserving rotational port `S`, which you can connect directly to a mechanical rotational port of a block from Simscape libraries. Therefore, for example, you can model the mechanical part of your system using Simscape and SimDriveline™ blocks, and then connect it directly to the electric drive through the mechanical rotational port `S`. For details, see the block reference pages.

Simscape Interface blocks now permitting user-defined initial conditions

There are several enhancements to the blocks in the Interface Elements library that let you connect SimPowerSystems and Simscape electrical circuits:

- Depending on the type of the interface block, you can specify the initial input and output voltage or current in the filter that is used to break an algebraic loop between SimPowerSystems and Simscape circuits. The Current-Voltage Simscape Interface and Current-Voltage Simscape Interface (gnd) blocks now have an additional **Initial Voltage** parameter. The Voltage-Current Simscape Interface and Voltage-Current Simscape Interface (gnd) blocks have an **Initial Current** parameter.

- The **Filter type** parameter in each of these blocks now has two additional values: `First-order input filtering` and `Second-order input filtering`. These options provide access to the input filtering functionality in the underlying Simulink-PS Converter block. For more information, see the Simulink-PS Converter block reference page.

- The block dialog boxes for each of these blocks now have an additional check box, **Show measurement ports**. If you select this check box and click **Apply** or **OK**, the block icon displays an additional port `m`. This is the output port of the underlying voltage or current sensor, depending on the type of the interface block. The Current-Voltage Simscape Interface and Current-Voltage Simscape Interface (gnd) blocks contain a current sensor. The Voltage-Current Simscape Interface and Voltage-Current Simscape Interface (gnd) blocks contain a voltage sensor. The output port `m` provides access to these measurements.

`power_fftscope` function now generating plots directly from MATLAB command prompt, with additional display options

You can now use the `power_fftscope` function directly at the command line, which provides the ability to automate the use of the tool. The THD computation has been improved to include the inter-harmonics in the THD calculations. Two new options are available in the **Display style** drop-down list:

- `Bar (relative to DC component)`
- `List (relative to DC component)`

For more information, see the power_fftscope reference page.

`power_lineparam` function accessible programmatically

You can now programmatically access a structure variable with default line geometry parameters, and use it as a template variable to configure a new line geometry and to compute RLC line parameters.

SimPowerSystems blocks supporting Simulink Parameter Objects as dialog box parameters

You can now use Simulink Parameter Objects when entering parameter values into the block dialog boxes. For more information, see Specify Parameter Values in the Simulink documentation.

R2012a

Version: 5.6

New Features

Bug Fixes

Simscape Rotational Port Available as Mechanical Input for Machine Blocks

The **Mechanical input** parameter has a new option, `Mechanical rotational port`, which has been added for the following machines:

- Simplified Synchronous Machine
- Synchronous Machine
- Asynchronous Machine
- Single Phase Asynchronous Machine
- DC Machine
- Permanent Magnet Synchronous Machine

When you select this option, the mechanical input port of the block (`Tm` or `w`), changes to a Simscape conserving rotational port `S`, which you can connect directly to a mechanical rotational port of a block from Simscape libraries. This allows you, for example, to model the mechanical part of your system using Simscape and SimDriveline blocks, and then connect it directly to the electrical machine through the mechanical rotational port `S`. See the block reference pages for details.

New Setup Function for Permanent Magnet Synchronous Machine Block

The power_PMSynchronousMachineParams function lets you compute parameters of a Permanent Magnet Synchronous Machine block based on standard manufacturer specifications. It lets you input manufacturer data and returns the computed machine parameters, along with additional derived data such as synchronous speed, number of pole pairs, nominal slip, starting torque, and so on. The function also lets you display relative errors between the input manufacturer data and the equivalent data obtained with the computed parameters. The function comes with a graphical user interface that allows you to compute the block parameters and apply it to selected block, as well as display the detailed results in the Command window.

Additional Solver Type Option Available for Discrete Mode

When you set the **Simulation type** parameter of the powergui block to `Discrete`, the dialog box now contains a new parameter, **Solver type**, which lets you select between `Tustin` and `Backward Euler` options. `Tustin` is the method used in previous versions. It is still the default, therefore there is no compatibility impact on existing models.

If your model shows numerical oscillations upon simulation in discrete mode, using `Backward Euler` method to discretize the state-space matrices may help eliminate the oscillations.

Distributed Resources Library Renamed to Renewable Energy

The Distributed Resources block library has been renamed, and is now called Renewable Energy library. All the block names are the same, and there is no compatibility impact on existing models.

New SimPowerSystems Demos

The following demos have been added in Version 5.6:

Demo Name	Description
Mechanical Coupling of Synchronous Generator with Exciter System Using the Simscape Mechanical Rotational Port (`power_SM_exciter_SSC`)	This is a modified version of the Mechanical Coupling of Synchronous Generator with Exciter System (`power_SM_exciter`) demo. It illustrates how using the new `Mechanical rotational port` option for mechanical coupling of an excitation system simplifies the model.
Initializing a 29-Bus, 7-Power Plant Network With the Load Flow Tool of Powergui (`power_LFnetwork_29bus`)	Illustrates the use of the Load Flow tool of powergui to initialize a 29-bus 735 kV network with detailed modeling of power plants using hydraulic turbines, speed regulation, excitation systems, and power system stabilizers.

R2011b

Version: 5.5

New Features

Bug Fixes

Compatibility Considerations

SimPowerSystems Software Now Requires Simscape Product

SimPowerSystems software now depends on and requires Simscape software, the foundation for Physical Modeling products. Simscape software includes common Physical Modeling utilities and block libraries. See the Simscape documentation for details.

SimPowerSystems block libraries are now located under the **Simscape** node in the Library Browser.

Interfacing with Simscape Electrical Domains

Four blocks in the new Interface Elements library allow you to connect a SimPowerSystems electrical connection line to a Physical Networks line connected to Simscape blocks. These blocks have SimPowerSystems ports on one side and Simscape ports on the other, and transfer voltage and current as Physical Networks Across and Through variables, respectively, without energy loss.

Sharing Models Using Simscape Editing Modes

SimPowerSystems software now features a selection of two Simscape editing modes that allow full or restricted editing of models.

- The Restricted mode requires SimPowerSystems product to be installed, but does not require a license. It allows you to change a limited set of model parameters, but not the blocks or connections, in a SimPowerSystems model.
- The Full mode requires SimPowerSystems product to be installed with a license. It allows you to change anything in a SimPowerSystems model.

For more details, see About the Simscape Editing Mode .

Block Library Links Must Be Resolved

All core SimPowerSystems blocks in your models must now have resolved block library links. You can neither disable nor break these library links. This is a global Simscape requirement. Consult the Simscape documentation for further details.

This restriction does not apply to the blocks from Application libraries (Electric Drives, Distributed Resources, and FACTS) and the Extra library.

Compatibility Considerations

If you have an existing SimPowerSystems model with disabled or broken links from the blocks in the model to the SimPowerSystems block libraries (other than the Applications and Extra library), you must restore all the broken block library links for your model to be valid.

If you have disabled or broken the SimPowerSystems library link for blocks that you have customized and want to keep these modified blocks in your model, you must move these modified blocks to your own custom library or libraries, then copy the block instances that you need to your model.

You must still restore the block link to its parent library, whether that parent is the SimPowerSystems block library or your own.

Changes to SimPowerSystems Demos

In Version 5.5 (R20011b), demos involving SimPowerSystems plus other Simscape products have been moved to the File Exchange (http://www.mathworks.com/matlabcentral/fileexchange). You can retrieve the demo models by following these links:

Demo Name	File Exchange Location
Fuel Cell Vehicle (FCV) Power Train (power_FCV_powertrain)	33309-fuel-cell-vehicle-fcv-power-train
Hybrid Electric Vehicle (HEV) Power Train (power_HEV_powertrain)	33310-hybrid-electric-vehicle-hev-power-train-using-battery-model
Electrically-Driven Hydraulic Motor Pump (power_Hydraulic_Pump)	33313-electrically-driven-hydraulic-motor-pump
Multi-Level Modeling for Rapid Prototyping of Complex Systems (power_HEV_MultiFidelity)	33315-multi-level-modeling-for-rapid-prototyping-of-complex-systems

R2011a

Version: 5.4

New Features

Bug Fixes

New Load Flow Tool

The **Load Flow and Machine Initialization** tool that was available in previous releases is renamed **Machine Initialization**. You can use it to set initial conditions of three-phase machines and regulators in order to start simulation in steady-state. The new **Load Flow** tool now provides an improved load flow solution for SimPowerSystems models, with several new features and capabilities:

· The load flow solution uses the Newton-Raphson method. It is more robust and provides a faster convergence than the **Machine Initialization** tool.

· The new **Load Flow** tool comes with a graphical user interface that allows you to display load flow solution at all buses.

· The Three-Phase Source and Three-Phase Programmable Voltage Source blocks are now taken into account in the load flow. You can now specify power and terminal voltage of the Three-Phase Source and Three-Phase Programmable Voltage Source blocks. Similar to the Synchronous Machine, you can declare these voltage sources as PV, PQ, or swing type. Once the load flow is solved, the source internal voltage magnitude and angle are automatically adjusted.

· You can now specify Three-Phase Series and Parallel RLC Load blocks either as constant impedance (constant Z) or as constant power (constant PQ). In the previous tool, the Three-Phase Series and Parallel RLC Load blocks were only considered as constant impedance loads based on the nominal voltage and active and reactive powers specified in the block menu.

The Load Flow Bus block has been added to the Measurements library. The power_loadflow function syntax has been modified, and a new **Load Flow** tab has been added to the powergui block parameters dialog box, as well as to the dialog boxes of all the blocks considered in the load flow solution. For more information on using the new **Load Flow** tool, see Load Flow Tool.

New Asynchronous Machine Block Setup Function

The power_AsynchronousMachineParams function lets you compute parameters of a double-cage Asynchronous Machine block based on standard manufacturer specifications. It lets you input manufacturer data and returns the computed machine parameters, along with additional derived data such as synchronous speed, number of pole pairs, nominal slip, starting torque, and so on. The function also lets you display relative errors between the input manufacturer data and the equivalent data obtained with the computed parameters. The function comes with a graphical user interface that allows

you to compute the block parameters and apply it to selected block, as well as display the detailed results in the Command window.

5-Phase Synchronous Machine Model Available

The Permanent Magnet Synchronous Machine block has been modified to allow modeling a 5-phase synchronous machine with sinusoidal back EMF and round rotor type. For details, see the block reference page.

SimState Support

SimPowerSystems software now supports Simulink SimState feature, introduced in R2009a. This feature allows you to save all runtime data necessary for restoring the simulation state of a model. For more information, see Saving and Restoring the Simulation State as the SimState in the *Simulink User's Guide*.

New SimPowerSystems Demo

The following demo has been added in Version 5.4:

Demo Name	Description
Flickermeter on a Distribution STATCOM (`power_flickermeter`)	Flickermeter model designed according to functional specifications of the international standard IEC 6100-4-15.

R2010b

Version: 5.3

New Features

Bug Fixes

Double Squirrel-Cage Rotor Option Available in Asynchronous Machine Block

The Asynchronous Machine block now lets you simulate a double squirrel-cage rotor. The **Rotor type** parameter in the block dialog box has a new option, `Double squirrel-cage`, in addition to the existing rotor modeling options of `Wound` and `Squirrel-cage`. For more information, see the block reference page.

Enhanced Code Generation Capabilities

Code generation support has been added for:

- Models in Phasor mode
- Models employing Ideal Switch mode (under Continuous)

For more information, see Improving Simulation Performance.

New SimPowerSystems Demos

The following demos have been added in Version 5.3:

Demo Name	Description
Synchronous Buck Converter (`power_switching_power_supply`)	Illustrates an abstracted version of a synchronous buck converter that uses ideal switching to give faster simulation times.
AC3 - Sensorless Field-Oriented Control Induction Motor Drive (`ac3_sensorless`)	Models a sensorless field-oriented control (FOC) induction motor drive with a braking chopper for a 200HP AC motor, using a modified version of the AC3 block.
AC7 - Sensorless Brushless DC Motor Drive During Speed Regulation (`ac7_sensorless`)	Models a sensorless brushless DC motor drive with a braking chopper for a 3HP motor, using a modified version of the AC7 block.
Electrically-Driven Hydraulic Motor Pump (`power_Hydraulic_Pump`)	Models a hydraulic pump driven by an electric motor, using SimPowerSystems and SimHydraulics® blocks. Note: This demo has been moved to the File Exchange in Version 5.5 (R20011b).

R2010a

Version: 5.2.1

New Features

Bug Fixes

Permanent Magnet Model Available in DC Machine Block

The DC Machine block now lets you model a permanent magnet DC machine, parameterized either by torque (torque per current constant) or by back-emf (voltage per speed constant). A new parameter, **Field type**, allows you to select between the wound-field and the permanent magnet DC machine. For more information, see the block reference page.

R2009b

Version: 5.2

New Features

Bug Fixes

Compatibility Considerations

Enhanced `power_cableparam` Function

The power_cableparam function, introduced in Version 5.1 (R2009a) as part of the Computation of R L and C Cable Parameters (`power_cable`) demo, is now available as a standalone command-line function with associated graphical user interface. It lets you compute RLC parameters of radial copper cables with single screen, based on conductor and insulator characteristics. For more information, see the power_cableparam reference page.

Changes to the Fuel Cell Stack Block

The Fuel Cell Stack block has been improved to better represent the cell dynamics. The model parameters and the meaning of some detailed parameters have changed since the last release, as described in the following section.

Compatibility Considerations

The Fuel Cell Stack block parameters have been changed in Version 5.2 (R2009b). If you used the **(No) User-Defined** option for the **Preset model** parameter in previous releases and defined particular detailed parameters for your Fuel Cell Stack block, the software will automatically convert your old block parameters into new values corresponding to the block changes.

The following table compares the old parameter names to the new ones. It also provides details on how the new values are computed:

Old Parameters	New Parameters and Values
Open circuit voltage	Voltage at 0A and 1A [V_0(V), V_1(V)] = [oldvalue, oldvalue*0.95]

New SimPowerSystems Demos

The following demos have been added in Version 5.2:

Demo Name	Description
Six-Pulse Cycloconverter (`power_cycloconverter`)	Illustrates a six-pulse cycloconverter driving a static load.

Speed Control of a DC Motor Using BJT H-Bridge (`power_Hbridge`)	Illustrates simulation of an H-bridge used to generate a chopped voltage and control speed of a DC motor, in open loop, in both directions.
Five-Cell Multilevel Converter (`power_fivecells`)	Illustrates a five-cell multilevel converter driving a static load.
Multi-Level Modeling for Rapid Prototyping of Complex Systems (`power_HEV_MultiFidelity`)	Illustrates how to use different detail level in model simulation. For more information, see Multi-Level Modeling for Rapid Prototyping. Note: This demo has been moved to the File Exchange in Version 5.5 (R20011b).

R2009a

Version: 5.1

New Features

Bug Fixes

Compatibility Considerations

Powergui Tools Are Also Available as Standalone Command-Line Functions

The graphical user interface analysis tools, available in the powergui block, are now also implemented as standalone command-line functions. Each of these tools can be activated by entering the appropriate command at the MATLAB prompt. For more information, see the following reference pages: power_fftscope, power_hysteresis, power_initstates, power_lineparam, power_loadflow, power_ltiview, power_report, power_steadystate, power_zmeter.

Enhancements to the Ideal Switching Algorithm

The **Display circuit differential equations** option, available in the powergui block parameters dialog box, lets you display differential equations of the model in the command window when the simulation starts. This option is visible only if **Enable use of ideal switching devices** is selected. For more information, see Using the Ideal Switching Device Method.

Powergui Block No Longer Added Automatically

The powergui block is no longer automatically added to your model upon simulation. You need to explicitly add it to your model. For more information, see Using the Powergui Block to Simulate SimPowerSystems Models.

Compatibility Considerations

If you have an old model without a powergui block, which used to run in previous releases because powergui was added automatically during simulation, you will now get an error trying to simulate it. Add a powergui block and save the model to avoid the error.

Changes to the Battery Block

The Battery block has been improved to accurately represent the battery dynamics during the charge and the discharge processes. The model parameters and the meaning of some detailed parameters have changed since the last release, as described in the following section.

Compatibility Considerations

The Battery block parameters have been changed in Version 5.1 (R2009a). If you used the **(No) User-Defined** option for the **Battery Type** parameter in previous releases and defined particular detailed parameters for your Battery block, the software will automatically convert your old block parameters into new values corresponding to the block changes.

The following table compares the old parameter names to the new ones. It also provides details on how the new values are computed:

Old Parameters	New Parameters and Values
Battery Type, set to `(No) User-Defined`	Battery Type, set by default to `Nickel-Metal-Hybrid`
-	Maximum Capacity (Ah) `= RatedCapacity*1.05`
Full charge voltage (%)	Fully charged voltage (v) `= [oldvalue]/100*NominalVoltage`
Nominal Discharge Current (% of Rated Capacity)	Nominal Discharge Current (A) `= [oldvalue]/100*RatedCapacity`
Capacity (% of Rated Capacity) @ Nominal Voltage	Capacity (Ah) @ Nominal Voltage `= [oldvalue]/100*RatedCapacity`

Exponential zone Voltage (%)
 	
Exponential zone Capacity
 	
(% of the Rated Capacity)
	Exponential zone Voltage (V)
	
	`= [oldvalue]/100*NominalVoltage`
	
	Exponential zone Capacity (Ah)
	
	`= [oldvalue]/100*RatedCapacity`

New SimPowerSystems Demos

The following demos have been added in Version 5.1:

Demo Name	Description
Computation of R L and C Cable Parameters (`power_cable`)	Illustrates use of the `power_cableparam` demo function to calculate the R,L, and C parameters of two 132kV 2-phase cables with screen cables.
Single-Phase Dynamic Load Block (`power_1phdynamicload`)	Illustrates an example of a single-phase dynamic load block built with Simulink blocks. This is the recommended template for interfacing custom-built Simulink blocks with SimPowerSystems models.
Zener Diode Regulator (`power_zener`)	Presents a model of the zener diode used in a voltage regulator.
Full-Wave Rectifier (`power_FullWaveRectifier`)	Illustrates use of the Ideal Switching Device solution method to simulate a full-wave rectifier using ideal diodes.
Multilevel Multiphase Space-Vector PWM (`power_svpwm_multiPhasesLevel`)	Illustrates modeling and operation of the Multilevel Multiphase Space-Vector PWM and Two-Level Multiphase Space-Vector PWM blocks. The demo includes a five-level five-phase inverter feeding a passive load.
Synchronous Generator and Full Scale Converter (Type 4) Detailed Model (`power_wind_type_4_det`)	Illustrates simulation of a 10 MW wind farm using a detailed model of a Type 4 wind turbine.
Synchronous Generator and Full Scale Converter (Type 4) Average Model (`power_wind_type_4_avg`)	Illustrates simulation of a 10 MW wind farm using an average model of a Type 4 wind turbine.

R2008b

Version: 5.0

New Features

Bug Fixes

New Ideal Switching Algorithm

New Ideal Switching algorithm, available in the powergui block, enables faster and more accurate simulation of power electronic devices. For more information, see Using the Ideal Switching Device Method.

Changes to the Universal Bridge Block

The Universal Bridge block has two new options for modeling voltage-sourced converters (VSC):

- Switching-function based VSC
- Average-model based VSC

New SimPowerSystems Demos

The following demos have been added in Version 5.0:

Demo Name	Description
Switching an Inductive Circuit Using a Breaker With no Snubber (power_breaker)	Illustrates the Ideal Switching device solution method of the powergui block.
Fuel Cell Vehicle (FCV) Power Train (power_FCV_powertrain)	Demonstration of a Fuel Cell Vehicle (FCV) power train using SimPowerSystems and SimDriveline. The FCV power train is of the series type. This FCV is propelled by one electric motor powered by a fuel cell and a battery. Note: This demo has been moved to the File Exchange in Version 5.5 (R20011b).

R2008a

Version: 4.6

New Features

Bug Fixes

Compatibility Considerations

New Fuel Cell Stack Block

A new block, Fuel Cell Stack, has been added to the Extra Sources sublibrary of the Electric Drives library. It implements a generic model parameterized to represent most popular types of fuel cell stacks fed with hydrogen and air. The 6 kW 45 Vdc Fuel Cell Stack demo (`power_fuel_cell`) shows how to use the Fuel Cell Stack block to model a Proton Exchange Membrane (PEM) Fuel Cell Stack feeding an average value 100Vdc DC/DC converter.

The Battery block from the Electrical Sources library is now included in the Extra Sources sublibrary of the Electric Drives library as well.

Initial Conditions Can Be Specified for the Permanent Magnet Synchronous Machine Block

The following enhancements have been implemented for the Permanent Magnet Synchronous Machine block:

- A new parameter, `Initial conditions`, allows you to specify the initial mechanical speed (rad/s), mechanical angle Θ (degrees) and instantaneous stator current (A).
- A new drop-down list lets you select the machine constant that you wish to specify for block parameterization: the flux linkage, the voltage constant, or the torque constant. Once you select a constant, you can enter its value in the appropriate parameter field, while the other two parameters become inaccessible and are only shown for information.
- The dialog box has been rearranged into three tabs, **Configuration**, **Parameters**, and **Advanced**, to improve usability.

Multiple Discretization Rates within a Model Now Available

For certain blocks, you can specify a different sample time than the one specified by the powergui block. This allows you to discretize different parts of a model at different rates in a fixed time step simulation. For example, if one block needs to run at a smaller time step (*t1*) than the rest of the simulation (*t2*), you can speed up simulation of the whole model by specifying a different time step for this block, as long as *t2* = *n* * *t1* (where *n* is an integer).

The following is a list of blocks that currently can be discretized at a different rate:

- Asynchronous Machine

- DC Machine
- Permanent Magnet Synchronous Machine
- Simplified Synchronous Machine
- Single Phase Asynchronous Machine
- Stepper Motor
- Switched Reluctance Motor
- Synchronous Machine

Dialog boxes for most of these blocks have also been rearranged into three tabs, **Configuration**, **Parameters**, and **Advanced**, to improve usability.

Compatibility Considerations

The DC Machine block can be discretized now. It is recommended that you use it instead of the Discrete DC Machine block, which will be deprecated in the future.

New SimPowerSystems Demos

The following demos have been added in Version 4.6:

Demo Name	Description
6 kW 45 Vdc Fuel Cell Stack (`power_fuel_cell`)	Demonstration of the Proton Exchange Membrane (PEM) Fuel Cell Stack model feeding an average value 100Vdc DC/DC converter. The nominal Fuel Cell Stack voltage is 45Vdc and the nominal power is 6kW.
Solid-Oxide Fuel Cell Connected to Three-Phase Electrical Power System (`power_SOFC`)	This demo illustrates a model of a solid oxide fuel cell (SOFC). The system consists of a SOFC, which is connected to a three-phase infinite bus through an IGBT inverter.
Mechanical Coupling of Synchronous Generator with Exciter System (`power_SM_exciter`)	In large alternators, the excitation system is provided by a small synchronous machine connected on the same shaft as the main synchronous machine. This demo illustrates interconnecting two machines on the same shaft by use of speed input.

Three-Phase Core-Type Transformer (`power_Transfo3phCoreType`)	This demo illustrates use of the Three-Phase Transformer Inductance Matrix Type block to model a three-phase core-type saturable transformer. It also demonstrates that using three single-phase transformers to simulate a Yg/Yg core-type transformer is not acceptable.
Three-Phase Matrix Converter (`power_three_phase_matrix_converter`)	This demo illustrates a three-phase matrix converter driving a static load. Indirect space-vector modulation allows direct control of input current and output voltage and hence allows the power factor of the source to be controlled. As a result, the demo outputs the unity power factor at the source.
Three-Phase Active Harmonic Filter (`power_active_filter`)	This demo illustrates the use of a shunt active harmonic filter (AHF) to minimize the harmonic content propagated to the source from a nonlinear load.

R2007b

Version: 4.5

New Features

Bug Fixes

Compatibility Considerations

New Battery Block

A new block, Battery, has been added to the Electrical Sources library. It implements a generic battery that models most popular battery types, such as Nickel-Metal-Hybride, Lead-Acid, Lithium-Ion, and Nickel-Cadmium. User-Defined Battery type allows you to modify detailed parameters to represent any particular discharge characteristics.

The Hybrid Electric Vehicle (HEV) Power Train demo (`power_HEV_powertrain`), which was introduced in Version 4.4 (R2007a) and shows a multi-domain simulation of a HEV power train based on SimPowerSystems and SimDriveline blocks, has been modified to use the Battery block. It is now called Hybrid Electric Vehicle (HEV) Power Train Using Battery Model.

New Stepper Motor Block

A new block, Stepper Motor, has been added to the Machines library. Depending on the motor configuration specified by the Motor type parameter, this block models:

- A two- or four-phase permanent magnet or hybrid stepper motor
- A three-, four-, or five-phase variable reluctance stepper motor

Three New Transformer Blocks

Three new transformer blocks have been added to the Elements library:

- Grounding Transformer implements a transformer that is used to provide a neutral in a three-phase, three-wire system. The transformer consists of three two-winding transformers connected in a zigzag. The nominal voltage of each of the six windings is Vn/3.
- Three-Phase Transformer Inductance Matrix Type (Two Windings) represents inductive coupling between windings located on different phases of a three-limb or a five-limb core. It also allows modeling of a three-phase transformer built with three single-phase units (no coupling between phases). The transformer R L parameters are obtained from no-load excitation tests and short-circuit tests in positive- and zero-sequence. When core type is specified as **Three-limb or five-limb core**, the transformer is modeled by 9 coupled windings; otherwise, it is modeled by 3 sets of 2 coupled windings (Z0=Z1).
- Three-Phase Transformer Inductance Matrix Type (Three Windings) represents coupling between windings located on different phases of a three-limb or a five-

limb core. It also allows modeling of a three-phase transformer built with three single-phase units (no coupling between phases). The transformer R L parameters are obtained from no-load excitation tests and short-circuit tests in positive- and zero-sequence. When core type is specified as **Three-limb or five-limb core**, the transformer is modeled by 9 coupled windings; otherwise, it is modeled by 3 sets of 3 coupled windings (Z0=Z1).

New Measurement Option Available for the PI Section Line Block

A new measurement option, `All pi-section voltages and currents`, is available for the PI Section Line block. It allows you to measure voltages and currents at the start and end of each pi-section.

New SimPowerSystems Demos

The following demos have been added in Version 4.5:

Demo Name	Description
Ni-MH Battery Model (power_battery)	Demonstration of the battery model during charge and discharge process. The demo models a 200 V, 6.5 Ah Ni-MH battery.
Stepper Motor Drive (power_steppermotor)	Demonstration of a hybrid stepper motor drive. The parameters are those of a small stepper motor (size 23).
D-STATCOM (Average Model) (power_dstatcom_avg)	In the average model of a Distribution STATCOM, the IGBT Voltage-Sourced Converters (VSC) are represented by equivalent voltage sources generating the AC voltage averaged over one cycle of the switching frequency. This model does not represent harmonics, but the dynamics resulting from the control system and power system interaction are preserved. This model allows using much larger time steps (typically 40-50 microseconds), thus allowing simulations of several seconds.
D-STATCOM (Detailed Model) (power_dstatcom_pwm)	The detailed model of a Distribution STATCOM includes detailed representation of power electronic IGBT converters. In order to achieve

	an acceptable accuracy with the 1680 Hz switching frequency used in this demo, the model must be discretized at a relatively small time step (5 microseconds). This model is well suited for observing harmonics and control system dynamic performance over relatively short periods of times (typically hundreds of milliseconds to one second).
UPFC (Detailed Model) (`power_upfc_gto48p`)	Detailed model of a 48-pulse, GTO-based Unified Power Flow Controller (500 kV, 100 MVA).

Renamed `psbhysteresis` Command

In Version 4.5 (R2007b), `power_hysteresis` is the new name for the old `psbhysteresis` command. You use it exactly the same way you would use the `psbhysteresis` command.

Compatibility Considerations

Currently, if you issue the `psbhysteresis` command, it will automatically redirect to its new name, `power_hysteresis`. However, it is recommended that you update your scripts and use the new command name going forward.

R2007a

Version: 4.4

New Features

Bug Fixes

Compatibility Considerations

New Brushless DC Motor Drive Block

A new block, Brushless DC Motor Drive, has been added to the Electric Drives/AC Drives library. It implements a brushless DC motor drive using a Permanent Magnet Synchronous Motor (PMSM) with trapezoidal back electromotive force (BEMF). It is possible to use a simplified version of the drive containing an average-value model of the inverter for faster simulation. In SimPowerSystems software, the Brushless DC Motor Drive block is commonly called the AC7 motor drive.

Automated Conversion of Version 2 Models Is No Longer Supported

The automated conversion of old models, created with blocks from SimPowerSystems 2.3 or Power System Blockset™ 2 libraries, is no longer supported in Version 4.4 (R2007a).

Compatibility Considerations

The psbupdate function is obsolete as of Version 4.4 (R2007a).

New SimPowerSystems Demos

The following demos have been added in Version 4.4:

Demo Name	Description
Hybrid Electric Vehicle (HEV) Power Train (power_HEV_powertrain)	Multi-domain simulation of a HEV power train based on SimPowerSystems and SimDriveline blocks. The HEV power train is of the series-parallel type, such as the one found in the Toyota Prius car. This HEV has two kinds of motive power sources, an electric motor and an internal combustion engine (ICE), in order to increase the drive train efficiency and reduce air pollution. It combines the advantages of the electric motor drive (no pollution and high available power at low speed) and the advantages of an internal combustion engine (high dynamic performances and low pollution at high speeds). Note: This demo has been moved to the File Exchange in Version 5.5 (R20011b).

Aircraft Electrical Power Generation and Distribution (`power_aircraft_distribution`)	This circuit illustrates a generic aircraft Electrical Power Generation & Distribution System. The AC power frequency is variable and depends of the engine speed.

R2006b

Version: 4.3

New Features

Bug Fixes

Mechanical Input Parameter Lets You Connect SimMechanics or SimDriveline Blocks to Electric Drives by Specifying Motor Speed as Block Input

The AC and DC Electric Drive blocks have a new parameter called **Mechanical input**, which lets you specify either the load torque or the motor speed as block input.

As of V4.3 (R2006b), mechanical input is available for the following blocks:

- Six-Step VSI Induction Motor Drive (AC1)
- Space Vector PWM VSI Induction Motor Drive (AC2)
- Field-Oriented Control Induction Motor Drive (AC3)
- DTC Induction Motor Drive (AC4)
- Self-Controlled Synchronous Motor Drive (AC5)
- PM Synchronous Motor Drive (AC6)
- Two-Quadrant Single-Phase Rectifier DC Drive (DC1)
- Four-Quadrant Single-Phase Rectifier DC Drive (DC2)
- Two-Quadrant Three-Phase Rectifier DC Drive (DC3)
- Four-Quadrant Three-Phase Rectifier DC Drive (DC4)
- One-Quadrant Chopper DC Drive (DC5)
- Two-Quadrant Chopper DC Drive (DC6)
- Four-Quadrant Chopper DC Drive (DC7)

To switch to the motor speed as mechanical input, open the block dialog box and set the **Mechanical input** option in the bottom portion of the dialog box to Speed w. Note that if you select the motor speed as mechanical input, the internal mechanical system is not used and the inertia and viscous friction parameters are not displayed. You have to include these parameters in the external mechanical system.

R2006a

Version: 4.2

New Features

Bug Fixes

Average Values of electricdrivelib Blocks

Two more AC drives blocks, AC3 and AC5, in the `electricdrivelib` library now have a new parameter that allows you to specify average-value models, as opposed to detailed models, for the converters. This parameter was first introduced for some of the `electricdrivelib` library blocks in V4.1 (R14SP2+).

As of V4.2 (R2006a), average-value models are available for the following blocks:

- Space Vector PWM VSI Induction Motor Drive (AC2)
- Field-Oriented Control Induction Motor Drive (AC3)
- Self-Controlled Synchronous Motor Drive (AC5)
- PM Synchronous Motor Drive (AC6)
- Two-Quadrant Single-Phase Rectifier DC Drive (DC1)
- Four-Quadrant Single-Phase Rectifier DC Drive (DC2)
- Two-Quadrant Three-Phase Rectifier DC Drive (DC3)
- Four-Quadrant Three-Phase Rectifier DC Drive (DC4)
- One-Quadrant Chopper DC Drive (DC5)
- Two-Quadrant Chopper DC Drive (DC6)
- Four-Quadrant Chopper DC Drive (DC7)

To switch to the average-value representation, open the block dialog box and set the **Model detail level** option in the bottom portion of the dialog box to `Average`.

Transformer Blocks with SI Units Are Available

The Transformer blocks now have a parameter named **Units**, which allows you to specify the `SI units` or the `pu units`. In addition, this parameter can be used to automatically convert pu units into SI units, or the reverse.

Open Circuit Option Is Added for the RLC Blocks

The RLC branch blocks how have an extra option under the **Branch Type** parameter that allows you to specify an `Open Circuit` branch. This is particularly useful if you want to temporarily get rid of an RLC element in the circuit without deleting the block.

New Demos and Enhancements to Existing Demos

The following demos have been added.

Demo Name	Description
`power_tcsc_phasor.mdl` `power_tcsc.mdl`	Thyristor Controlled Series Capacitor (TCSC) test systems from Dr. Dragan Jovcic from University of Aberdeen, UK
`power_asm1ph_auxcontrol` `power_asm1ph_vectorcontrol`	The single-phase Asynchronous Machine block using the `Main & auxiliary windings` configuration

In the Wind Farm DFIG demos of the Distributed Resources library, control systems have been enhanced.

R14SP3

Version: 4.1.1

Bug Fixes

R14SP2+

Version: 4.1

New Features

Bug Fixes

Compatibility Considerations

New Blocks in the Machines Library of powerlib

The machines library of powerlib contains a model of a Switched Reluctance Motor that allow you to model three typical configurations of such a motor. The library also contains a model of a Single-Phase Asynchronous Machine that can be configured as a split phase, a capacitor-start, or as a capacitor-start-run motor mode.

Enhancements to Existing Blocks of the Machines Library

The core saturation can now be specified for the Asynchronous Machine block when the block is used in a phasor simulation. The saturation model of the Asynchronous Machine is based on the fundamental component of the current and does not include the third harmonic. The saturation parameter is available only when the simulation is in phasor mode. When the powergui block is set to continuous or discrete mode, the parameter is disabled in the mask of the block.

The Permanent Magnet Synchronous Machine block allows you to specify a trapezoidal flux distribution as an alternative to the sinusoidal flux option of the previous version of the block.

Branch Type Parameter of the RLC Branch Blocks

The Series RLC Branch block, Parallel RLC Branch block, Three-Phase Series RLC Branch block, and Three-Phase Parallel RLC Branch block now have a new parameter that allows you to directly specify the elements that are present in the branch: the R, L, C, RL, LC, RC, and RLC configurations can be specified. It is no longer required to specify an Inf value for the Capacitance in a Series RLC Branch block to get rid of the capacitor device in the branch or to specify 0 value of resistance to get rid of the resistor of a Parallel RLC Branch block.

Average Values of electricdrivelib Blocks

The seven DC drives blocks, and the AC2 and AC6 AC drives of the `electricdrivelib` library now have a new parameter that allows you to specify average value models, as opposed to detailed models, for the converters.

To switch to the average-value representation, open the block dialog box and set the **Model detail level** option in the bottom portion of the dialog box to `Average`.

Obsolete Blocks

The Discrete System block and the Machine Measurement Demux block are no longer supported in V4.1 (R14SP2+).

Compatibility Considerations

The table below indicates blocks that are obsolete as of the current version, and lists blocks that you can use as replacement for the obsolete blocks.

Obsolete Block	Removed from Version	Replacement
Discrete System	4.1	powergui
Machine Measurement Demux	4.1	Bus Selector

R14SP2

Version: 4.0.1

Bug Fixes