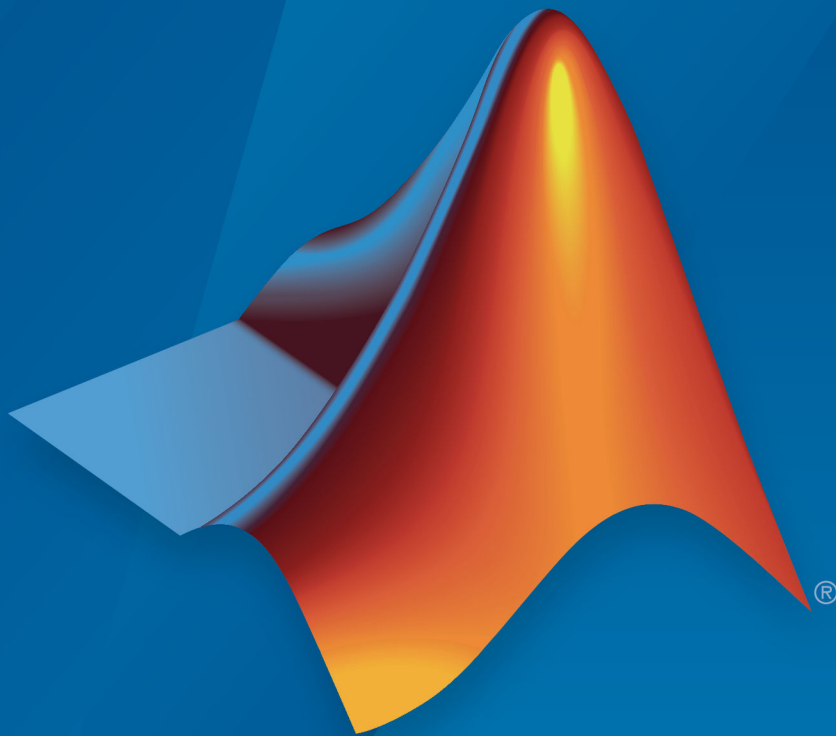


Simscape™ Electronics™ Release Notes



MATLAB® & SIMULINK®

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

Simscape™ Electronics™ Release Notes

© COPYRIGHT 2008–2017 by The MathWorks, Inc.

The software described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from The MathWorks, Inc.

FEDERAL ACQUISITION: This provision applies to all acquisitions of the Program and Documentation by, for, or through the federal government of the United States. By accepting delivery of the Program or Documentation, the government hereby agrees that this software or documentation qualifies as commercial computer software or commercial computer software documentation as such terms are used or defined in FAR 12.212, DFARS Part 227.72, and DFARS 252.227-7014. Accordingly, the terms and conditions of this Agreement and only those rights specified in this Agreement, shall pertain to and govern the use, modification, reproduction, release, performance, display, and disclosure of the Program and Documentation by the federal government (or other entity acquiring for or through the federal government) and shall supersede any conflicting contractual terms or conditions. If this License fails to meet the government's needs or is inconsistent in any respect with federal procurement law, the government agrees to return the Program and Documentation, unused, to The MathWorks, Inc.

Trademarks

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See www.mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

Patents

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

Semiconductors Characteristics Visualization: Plot I-V curve for semiconductor device model parameter values	1-2
Passive Device Blocks: Model slow-blow fuse, diffusion resistor, and reluctance with hysteresis	1-2
RC Servo Block: Model fault-capable RC servo systems	1-3
Gate Driver and Half-Bridge Driver Blocks: Model gate driver integrated circuits	1-3
Finite-Gain Op-Amp Block with Noise: Model thermal noise effects	1-4
Controlled PWM Voltage Block Enhancements: Select between electrical and physical signal input ports and specify minimum pulse duration	1-4
N-Channel IGBT with Tabulated Thermal Dependence: Use thermal-dependent lookup table approximation to I-V curve	1-5
Thermocouple with Type K Support: Model K-series thermocouples	1-5
Variable Target Specification for Additional Blocks: Guide initialization using desired variable values with relative priority	1-6
Relay Block Improved Usability: Set block initial state based on input	1-6

SPICE Semiconductor Parameter Renaming: Make block parameter names consistent with SPICE2	1-7
New examples	1-7

R2017a

MOSFET Characteristics Viewer: Verify model behavior based on specified parameter values	2-2
Battery Fade: Model age-related degradation of battery performance	2-2
Transformer Block with Fault Parameterization: Test system behavior during faults	2-3
elec_getPowerLossTimeSeries and elec_getEfficiency Functions: Calculate time-series power loss data and total efficiency for the model	2-3
elec_calculateFluxPartialDerivatives Function: Convert FEM-exported flux linkage data to parameters used by FEM-Parameterized PMSM block	2-4
MOSFET Block with Surface Scattering and Tabulated Capacitances: Model reduced electron mobility due to electric field strength	2-4
FEM-Based PMSM with Simplified Inductance Parameterization: Specify rotor inductances independent of rotor angle	2-5
Variable Target Specification: Guide initialization using desired variable values with relative priority	2-6
Table-Based N-Channel IGBT block: Changed extrapolation behavior	2-6
New examples	2-7

MOSFET Block with Surface Potential Model: Model accurate behavior near threshold voltage while conserving charge 3-2

LDMOS Transistor Block: Model LDMOS transistors suitable for high voltage, including quasisaturation 3-3

Generic Battery Block Thermal and Measurement Ports: Model heat transfer from battery and access internal charge measurement 3-3

Faults and Tabulated Efficiencies for Servomotor Block: Inject faults into servomotor and use tabulated efficiencies 3-4

FEM-Parameterized PMSM Block Enhancements: Use 4D flux data, Park transformation for 3D data, and improved iron losses model 3-5

New examples 3-6

SimElectronics renamed to Simscape Electronics 4-2

Capacitor Block and Inductor Block with Fault Parameterization: Test system behavior during faults 4-2

Thermal Resistor block now part of Simscape Foundation library 4-2

Parameters change to N-Channel IGBT, N-Channel MOSFET, and P-Channel MOSFET blocks 4-3

New examples 4-3

Fault block for modeling open-circuit and short-circuit electrical faults	5-2
Operating limits, tolerances, and fault parameterization for Resistor block	5-2
FEM-Parameterized PMSM block	5-3
Servomotor block with tabulated efficiency option	5-3

elec_getPowerLossSummary function to calculate and view semiconductor switching device losses	6-2
Nonlinear magnetization inductance with hysteresis option for Nonlinear Inductor and Nonlinear Transformer blocks	6-2
Schmitt Trigger block	6-3
Current Limiter block	6-3
Droop parameterization for DC-DC Converter block	6-3
Thermal port available for H-Bridge block	6-4
New examples	6-4

Averaged mode for stepper motor and driver blocks	7-2
Torque or force calculation option for FEM actuator blocks	7-2
Nonzero threshold width in Switch blocks	7-2
Event-based variant of the N-Channel IGBT block	7-3
Tabulated capacitance options for MOSFET and IGBT blocks	7-4
Z pulse width change in Incremental Shaft Encoder block	7-4
New examples	7-5

Unipolar Stepper Motor Driver block and half-stepping option for stepper motor drivers	8-2
Detent torque for stepper motors	8-2
Output capacitance for MOSFET and IGBT blocks	8-2
Lookup table definition of I-V relationship in IGBT and Thyristor blocks	8-3
Thyristor block that now models devices with higher leakage currents	8-3
New examples	8-3

R2013b

Voltage-Controlled Oscillator block	9-2
Servomotor block that now permits parameterized torque-speed curve and optional iron losses	9-2
Solar Cell block that now models solar cell arrays	9-2
New examples	9-3

R2013a

Resistor with optional thermal noise	10-2
Voltage and current sources with DC, AC, and noise components	10-2
Nonlinear charge model for N-Channel MOSFET, P-Channel MOSFET, and N-Channel IGBT blocks	10-3
New examples	10-3

R2012b

Single-pole and double-pole switch blocks	11-2
Five new sensor blocks: PTC Thermistor, Pressure Transducer, Resolver, Gyro, and Accelerometer	11-2
Nonlinear Transformer and Nonlinear Inductor blocks ...	11-3
DC-DC Converter block and Incandescent Lamp block	11-3

Unipolar Stepper Motor block and Stepper Motor block that now account for iron losses	11-4
New examples	11-4

R2012a

Thermal Ports Available for Actuator Blocks and for Solar Cell Block	12-2
Fully Differential Op-Amp Block	12-2
Transmission Line Block	12-2
Power Sensor Block	12-2
Induction Motor Block Usability Enhancement	12-3
Controlled PWM Voltage Block Enhancement	12-3
Exponential Diode Block Now Models Charge Dynamics ..	12-3
New Demos	12-3

R2011b

Thermal Ports Available for Semiconductor Blocks	13-2
New Operational Transconductance Amplifier Block	13-2
New Push-Pull Output Block	13-2
H-Bridge Block Enhancements	13-3

DC Motor Block Supports No-Load Current Data for Rotor Damping	13-3
New Demos	13-4

R2011a

Thermal Dependency Added to Semiconductor Blocks	14-2
New Demos	14-2

R2010b

New Thyristor Block	15-2
New Multiplier Block	15-2
Additional Exponential Diode Parameterization Options	15-2
Channel Modulation Parameter Added for MOSFET Blocks	15-2
Changes to the Bipolar Transistor Blocks	15-2
New Demos	15-3

R2010a

New Linear and Rotary Motors Defined in Terms of Flux	16-2
--	-------------

New Potentiometer Block	16-2
Initial Conditions Tab Added for Logic Blocks	16-2
Changes in Block Parameterization	16-2
Changes to the SPICE-Compatible Blocks	16-2
Changes to the Solar Cell Block	16-3
New Demos	16-3
Functions and Function Elements Being Removed	16-4

R2009b

Actuators & Drivers Library Blocks	17-2
New Generic Rotary Actuator Block	17-2
New Generic Linear Actuator Block	17-2
Improved Servomotor Block	17-2
New Abstracted Timer Block	17-2
New Demos	17-2

R2009a

Actuators & Drivers Library	18-2
New Piezo Motor Blocks	18-2
Enhanced H-Bridge Block	18-2
Passive Devices Library	18-2
New Switch Blocks	18-2
New Resistor Block	18-3
New Crystal Block	18-3
Enhanced Variable Inductor and Variable Capacitor Blocks	18-3

Sources Library	18-3
Enhanced Solar Cell Block	18-3
New Two-Input Dependent Source Blocks	18-3
SPICE-Compatible Semiconductors Library	18-4
Enhanced NMOS and PMOS Blocks	18-4

R2008b

New CMOS Logic Gate Blocks	19-2
New Piezo Stack Block	19-2
New Relay Block	19-2
New Fuse Block	19-2
New NMOS and PMOS Blocks	19-2

R2008a+

Product Introduction	20-2
-----------------------------------	-------------

R2017b

Version: 2.12

New Features

Compatibility Considerations

Semiconductors Characteristics Visualization: Plot I-V curve for semiconductor device model parameter values

A new quick plot feature lets you visualize the basic I-V characteristics for semiconductor switching devices. To plot the characteristics, right-click a semiconductor block in your model and, from the context menu, select **Electronics > Basic characteristics**. The software automatically computes a set of bias conditions, based on the block parameter values, and opens a figure window containing a plot of the DC I-V characteristics for the block.

This feature is implemented for nonthermal variants of the following blocks in the Semiconductors library:

- N-Channel IGBT
- N-Channel MOSFET (both threshold-based and surface-potential-based variants)
- P-Channel MOSFET (both threshold-based and surface-potential-based variants)
- N-Channel LDMOS FET
- P-Channel LDMOS FET
- N-Channel JFET
- P-Channel JFET
- NPN Bipolar Transistor
- PNP Bipolar Transistor

Also, the MOSFET Characteristics Viewer, which was introduced in R2017a for surface-potential-based N-Channel MOSFET blocks, has now been extended to include surface-potential-based P-Channel MOSFET blocks. This tool lets you perform an in-depth study of block characteristics and match the block behavior to a set of target characteristics. To access the Characteristics Viewer tool, right-click a surface-potential-based MOSFET block in your model and, from the context menu, select **Electronics > Explore characteristics**.

Passive Device Blocks: Model slow-blow fuse, diffusion resistor, and reluctance with hysteresis

The new Reluctance with Hysteresis block in the Passive Devices library lets you model the physical effects of a nonlinear reluctance with magnetic hysteresis. Use this block as

a building block for custom inductances and transformers that exhibit magnetic hysteresis.

The new Diffusion Resistor block in the Passive Devices library represents a resistor with velocity saturation. Similarly to the existing Resistor block, this block also lets you model tolerances, operational limits, fault behavior, and thermal noise. You can turn these modeling options on and off independently of each other.

The Fuse block in the Passive Devices library now has additional parameterization options that let you model dependence of opening time on current:

- Opening time is independent of current — This option implicitly assumes that heat in the fuse is instantaneously dissipated once the current drops below a certain threshold. It also explicitly assumes that the time to open is independent of the current. This is the same functionality as in previous releases.
- Opening time depends on $I^2 \cdot t$ — Makes fuse opening dependent on the energy dissipated in the fuse. The nominal melting value ($I^2 \cdot t$) is usually available on manufacturer datasheets. Use this option for high-current range of operation.
- Opening time is tabulated — Makes fuse opening dependent on the average time-current tabulated data. This data indicates the time that it takes for the fuse to open given a certain current, and is available on manufacturer datasheets as time-current curves. Use this option for small currents.

RC Servo Block: Model fault-capable RC servo systems

The new RC Servo block in the Rotational Actuators library represents a small DC motor with a gearbox and control circuitry, commonly used in quadcopters, radio-controlled planes and helicopters, and other mechatronic devices. RC servos provide angular position control of the output shaft over a limited angle range. The angle demand is set by the pulse width of a PWM signal applied to port s.

Gate Driver and Half-Bridge Driver Blocks: Model gate driver integrated circuits

Two blocks have been added to the Drivers library:

- Gate Driver — Provides an abstracted representation of a gate driver integrated circuit.

- **Half-Bridge Driver** — Provides an abstracted representation of an integrated circuit for driving MOSFET and IGBT half-bridges.

Both blocks model input hysteresis, propagation delay, and turn-on/turn-off dynamics.

Unless modeling a gate driver circuit explicitly, always use one of these blocks to set gate-source voltage on a MOSFET block or gate-emitter voltage on an IGBT block. Do not connect a controlled voltage source directly to a semiconductor gate, because this omits the gate driver output impedance that determines switching dynamics.

Finite-Gain Op-Amp Block with Noise: Model thermal noise effects

The Finite-Gain Op-Amp block can now generate thermal noise voltage and current. The equivalent circuit contains two noise current sources, attached to the two block inputs, and a noise voltage source attached to the noninverting input. These three sources are independent from each other.

The default value of the new **Noise mode** parameter is `Disabled`, in which case the block does not generate noise. If you select `Enabled`, additional parameters become available in the **Noise** section of the block interface. By default, the same current density is applied to both inputs, but you can apply different current densities to each input by using the **Noise current parameterization** parameter. The block generates Gaussian noise by using the Random Number source block in the Simscape Foundation library. The **Repeatability** parameter gives you the following options for noise control: not repeatable, repeatable with a random seed, and repeatable with explicitly specified seed.

Controlled PWM Voltage Block Enhancements: Select between electrical and physical signal input ports and specify minimum pulse duration

The Controlled PWM Voltage block now has two modeling variants, accessible by right-clicking the block in your block diagram and then selecting the appropriate option from the context menu, under **Simscape > Block choices**:

- **Electrical input ports** — The block calculates the duty cycle based on the reference voltage across its +ref and -ref ports. This modeling variant is the same as in previous releases.
- **Physical signal input** — Specify the duty cycle value directly by using an input physical signal port.

Both block variants have a new parameter, **Minimum pulse width**, which lets you specify the minimum pulse duration.

N-Channel IGBT with Tabulated Thermal Dependence: Use thermal-dependent lookup table approximation to I-V curve

The N-Channel IGBT block now has an additional parameterization option. In previous releases, the detailed block variant gave you a choice of using either the equivalent circuit representation or a lookup table. However, the second option used a 2-D table lookup that did not include temperature dependence data.

Now the **Parameterization** parameter for the detailed block variant has three options:

- `Fundamental nonlinear equations` — Use an equivalent circuit based on a PNP bipolar transistor and N-channel MOSFET (same as in previous releases).
- `Lookup table (2-D, temperature independent)` — Use 2-D table lookup for collector current as a function of gate-emitter voltage and collector-emitter voltage (same as `Lookup table` in previous releases).
- `Lookup table (3-D, temperature dependent)` — Use 3-D table lookup for collector current as a function of gate-emitter voltage, collector-emitter voltage, and temperature.

If you use the `Lookup table (3-D, temperature dependent)` option and the block thermal port is not exposed, then the **Device simulation temperature** parameter on the **Temperature Dependence** tab lets you specify the simulation temperature.

Thermocouple with Type K Support: Model K-series thermocouples

The Thermocouple block now lets you model NIST Type K thermocouples. Use the new **Thermocouple type** parameter to select the thermocouple type:

- `Type B, E, J, K (t<=0 degC), N, R, S or T` — This option is equivalent to the block functionality in previous releases.
- `Type K (t>=0 degC)` — This option adds an exponential term to the block equations when the temperature difference is greater than 0 degrees Celsius.

Variable Target Specification for Additional Blocks: Guide initialization using desired variable values with relative priority

In R2017a, block-level variable initialization was introduced for several Simscape Electronics blocks (see “Variable Target Specification: Guide initialization using desired variable values with relative priority” on page 2-6).

In this release, block-level variable initialization is also available for the following blocks:

- Accelerometer
- Gyro
- PTC Thermistor
- Pressure Transducer
- Resolver
- Thermistor

The values you specify during block-level variable initialization are not the actual values of the respective variables, but rather their target values at the beginning of simulation ($t = 0$). Depending on the results of the solve, some of these targets may or may not be satisfied. For more information, see “Variable Initialization” (Simscape).

Relay Block Improved Usability: Set block initial state based on input

The Relay block no longer has the **Initial connection** parameter, which used to specify whether the relay was open or closed at the start of simulation.

Now the initial state of the block is set automatically based on the signal value at port PS at the start of simulation ($t = 0$):

- If PS is greater than the threshold, the common port C connects to the S2 contact (relay is closed).
- If PS is less than or equal to the threshold, the common port C connects to the S1 contact (relay is open).

Compatibility Considerations

In legacy models, if the parameter value was set correctly based on the expected input at $t = 0$, there is no change in simulation results. Otherwise, the initial state of the relay is flipped.

SPICE Semiconductor Parameter Renaming: Make block parameter names consistent with SPICE2

Several blocks in the Additional Components library have had their parameters renamed, to make them consistent with SPICE2.

- SPICE Diode block:
 - **ND** renamed to **N**
 - **MG** renamed to **M**
- SPICE NJFET and SPICE PJFET blocks:
 - **ND** renamed to **N**
 - **MG** renamed to **M**
- SPICE NMOS and SPICE PMOS blocks:
 - **ND** renamed to **N**

Compatibility Considerations

Blocks in existing models will continue to work. If you have scripts that use `get_param` and `set_param` and reference the old parameter names, then these scripts will generate errors. Update these scripts to use the new parameter names.

New examples

Examples introduced in this version are:

- Custom Solenoid with Magnetic Hysteresis
- Electrical Transformer with Hysteresis

R2017a

Version: 2.11

New Features

Compatibility Considerations

MOSFET Characteristics Viewer: Verify model behavior based on specified parameter values

The new MOSFET Characteristics Viewer tool allows you to study characteristics of a particular parameterization of a surface-potential-based MOSFET block and match the block behavior to a set of target characteristics. The tool allows you to:

- Plot simulated data, using the current block parameters.
- Overlay simulated data plots over tabulated target data.
- Modify block parameters.
- When satisfied with the results of the parameters tuning in the Characteristics viewer, update the block parameters in the model.
- Save generated parameter sets for future reuse in a different model.

For more information, see MOSFET Characteristics Viewer.

Battery Fade: Model age-related degradation of battery performance

The Battery block (formerly called Generic Battery) now contains a new parameter, **Model battery fade?** Setting it to `Include` enables you to specify the dependence of other battery parameters on the charge-discharge history.

This 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, you can simulate a vehicle transmission system assuming a new battery, or a battery that requires replacing, without modifying multiple model parameters.

See the block reference page for details.

Compatibility Considerations

The block name has changed from Generic Battery to Battery. The name of the Simscape language source file has changed accordingly.

If you are using the block in your models, there is no compatibility impact because the block will be updated automatically when you open your model.

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

Transformer Block with Fault Parameterization: Test system behavior during faults

The new Mutual Inductor block in the Passive Devices library lets you model a mutual inductor (two-winding transformer) with nominal inductance tolerances for each winding, operating limits, and faults.

Transformers are typically rated with a particular saturation current, and sometimes with a maximum power dissipation. You can use these values as operating limits to generate warnings or errors if the block is driven outside of its specification.

Fault modeling options are the same as for the Inductor block:

- At a specific time
- When the maximum voltage is exceeded more than a certain number of times
- When the maximum current is exceeded for longer than a specific time interval

Either the primary or the secondary winding can be faulted. The faulted winding is modeled as a faulted inductor. The unfaulted winding is coupled to the faulted winding.

elec_getPowerLossTimeSeries and elec_getEfficiency Functions: Calculate time-series power loss data and total efficiency for the model

Checking dissipated power is useful for verifying that circuit components are operating within their working envelopes. In addition to the blocks in the Semiconductor Devices library, some other blocks now also have an internal variable called `power_dissipated`, which represents the instantaneous power dissipated by the component. When you log simulation data, the time-value series for this variable represents the power dissipated by the component over time. You can view and plot this data using the Simscape Results Explorer.

The `elec_getPowerLossSummary` function, added in R2015a, lets you calculate average losses for a block over a period of time. To improve usability, the output table of this function is now sorted with respect to the power.

Two new functions extend the reporting on switching device losses:

- The `elec_getPowerLossTimeSeries` function calculates dissipated power losses for blocks in a model, based on logged simulation data, and returns the time series data for each block.
- The `elec_getEfficiency` function calculates efficiency for blocks in a model, based on logged simulation data.

To use these functions, you have to enable simulation data logging and run the simulation. For more information, see [Data Logging](#).

elec_calculateFluxPartialDerivatives Function: Convert FEM-exported flux linkage data to parameters used by FEM-Parameterized PMSM block

The FEM-Parameterized PMSM block works with flux linkage partial derivatives, rather than directly with flux linkage, for improved numerical performance.

If your finite-element motor design tool does not have an option to output partial derivatives, then you can use the new `elec_calculateFluxPartialDerivatives` function to calculate these derivatives from the flux linkage.

You can also use the new `elec_generateIdealPMSMfluxData` function to generate tabulated flux linkage data for an ideal permanent magnet synchronous motor. Use this function to create test data for the FEM-Parameterized PMSM block, either for validation purposes or to set up a model before actual flux linkage data is available.

MOSFET Block with Surface Scattering and Tabulated Capacitances: Model reduced electron mobility due to electric field strength

The N-Channel MOSFET and P-Channel MOSFET blocks have the following enhancements in this release:

- The surface-potential variants of these blocks now let you model surface scattering. The **Main** section for the surface-potential variants of the block interface contains a new parameter, **Surface roughness scattering factor**, which indicates the strength of the mobility reduction due to surface scattering. If you model temperature dependence, the new **Surface roughness scattering temperature exponent**

parameter leads to a temperature-dependent reduction in the MOSFET transconductance at high gate voltage.

- For threshold-based variants, the tabulated capacitance models have an improved voltage dependence control of the reverse transfer capacitance, C_{rss} , and the gate-drain junction capacitance. These capacitances are now a function of the drain-gate voltage, which provides better modeling for the gate-bias dependence of the input capacitance. The new parameter, **Gate-source voltage, V_{gs} , for tabulated capacitances**, is available in the **Junction Capacitance** section of the block interface. This parameter is visible only for tabulated capacitance models (Specify tabulated input, reverse transfer and output capacitance or Specify tabulated gate-source, gate-drain and output capacitance). The block calculates drain-gate voltages by subtracting this gate-source voltage value from the values specified for the **Corresponding drain-source voltages** parameter (N-Channel MOSFET), or from the negative of the values specified for the **Corresponding source-drain voltages** parameter (P-Channel MOSFET). The default **Gate-source voltage, V_{gs} , for tabulated capacitances** parameter value is 0 V.

To obtain the full Miller capacitance behavior, it is preferable to include bias conditions covering both negative and positive drain-gate voltages. However, this information is not always available from datasheets. Depending on the application, you might get better results by using a fixed capacitance parameterization and setting the **Charge-voltage linearity** parameter to Gate-drain charge function is nonlinear. This is particularly true for sensitive gate-driving circuitry.

Compatibility Considerations

To avoid an incompatibility, MOSFET blocks in models created prior to R2017a have the **Gate-source voltage, V_{gs} , for tabulated capacitances** parameter value of NaN (not a number), and the capacitance continues to be a function of drain-source (or source-drain) voltages. The NaN value is intended only to preserve the current behavior of existing models. For new models, set this parameter to a numeric value (the default value is 0 V).

FEM-Based PMSM with Simplified Inductance Parameterization: Specify rotor inductances independent of rotor angle

The FEM-Parameterized PMSM block now lets you specify tabulated data for d -axis and q -axis inductances (L_d and L_q) as a function of d -axis and q -axis currents, instead of

phase current and rotor angle. The 2-D Data block variant now has an additional parameter, **Parameterization**, with the following options:

- Assume constant mutual inductance - tabulate with phase current and rotor angle — The 2-D parameterization option available in previous releases.
- Assume sinusoidal back emf - tabulate with d- and q-axis currents — The new option, which assumes that the flux linking each winding depends nonlinearly on all stator winding currents. It also assumes that the permanent magnet flux linkage is sinusoidal. This option is usually a good fit for interior magnet PMSMs (or IPMSMs).

Variable Target Specification: Guide initialization using desired variable values with relative priority

At the beginning of simulation, the solver computes the initial conditions to determine the simulation starting point. You can affect the initial conditions computation by block-level variable initialization, that is, by specifying the priority and target initial values for certain variables in the **Variables** section of the block dialog.

In this release, block-level variable initialization is available for the following blocks:

- Capacitor
- Inductor
- Diode
- Light-Emitting Diode
- Photodiode
- Optocoupler

The values you specify during block-level variable initialization are not the actual values of the respective variables, but rather their target values at the beginning of simulation ($t = 0$). Depending on the results of the solve, some of these targets may or may not be satisfied. For more information, see Variable Initialization.

Table-Based N-Channel IGBT block: Changed extrapolation behavior

In previous releases, if the applied voltage became lower than the lowest gate voltage specified by the **Vector of gate-emitter voltages**, **Vge** parameter, the block performed

linear extrapolation. In some cases, this could lead to negative power dissipation in the block, with the collector current opposing the applied collector-emitter voltage.

This incorrect behavior is no longer possible. In this release, if the applied voltage is outside the defined range of tabulated data and the extrapolation results in the incorrect current direction, the block clips the collector current to zero and issues a warning. You can then modify the tabulated values, so that the applied voltages are within the table and the extrapolation is not needed.

In general, the best practice is to define the tabulated collector-current values in such a way that dI_c/dV_{ce} and dI_c/dV_{ge} go to zero along the low-bias boundaries of the tabulated data.

Compatibility Considerations

If your legacy model used out-of-range values of applied voltage, your simulation results can change. If you get a warning, modify the tabulated values, so that the applied voltages are within the table range and the extrapolation is not needed.

New examples

Examples introduced in this version are:

- Energy Scavenger
- MOSFET Fault in Buck Converter
- Stepper Motor Averaged Mode

R2016b

Version: 2.10

New Features

Compatibility Considerations

MOSFET Block with Surface Potential Model: Model accurate behavior near threshold voltage while conserving charge

In previous releases, the N-Channel MOSFET and P-Channel MOSFET blocks used the Shichman-Hodges equation to represent the device. This modeling approach, based on threshold voltage, has the benefits of simple parameterization and simple current-voltage expressions. However, these models have difficulty in accurately capturing transitions across the threshold voltage and lack some important effects, such as velocity saturation.

Now, both the N-Channel MOSFET and P-Channel MOSFET blocks also have variants that model a MOSFET using the surface potential. To access these block variants, right-click the block in your block diagram and then select the appropriate option from the context menu, under **Simscape > Block choices**:

- **Threshold-based** — Basic model, which represents the device using the Shichman-Hodges equation (based on threshold voltage) and does not simulate thermal effects. This is the default. This model is equivalent to the default (nonthermal) block variant available in previous releases.
- **Threshold-based with thermal** — Model based on threshold voltage and with exposed thermal port. This model is equivalent to the thermal block variant available in previous releases.
- **Surface-potential-based** — Model based on surface potential, new for R2016b. This model does not simulate thermal effects.
- **Surface-potential-based with thermal** — Thermal variant of the model based on surface potential, new for R2016b.

The surface-potential-based MOSFET models for n-channel and p-channel devices provide a greater level of model fidelity than the simple square-law (threshold-voltage-based) models. The surface-potential-based block variants include the following effects:

- Fully nonlinear capacitance models (including the nonlinear Miller capacitance)
- Charge conservation inside the model, so you can use the model for charge sensitive simulations
- Velocity saturation and channel-length modulation
- The intrinsic body diode
- Reverse recovery in the body diode model

-
- Temperature scaling of physical parameters
 - For the thermal variants, dynamic self-heating (that is, the effect of self-heating on the electrical characteristics of the device can be simulated)

LDMOS Transistor Block: Model LDMOS transistors suitable for high voltage, including quasisaturation

Two new blocks in the Semiconductors library, N-Channel LDMOS FET and P-Channel LDMOS FET, let you model LDMOS (or VDMOS) transistors suitable for high voltage. The models are based on surface potential and include effects due to an extended drain (drift) region:

- Nonlinear capacitive effects associated with the drift region
- Surface scattering and velocity saturation in the drift region
- Velocity saturation and channel-length modulation in the channel region
- Charge conservation inside the model, so you can use the model for charge sensitive simulations
- The intrinsic body diode
- Reverse recovery in the body diode model
- Temperature scaling of physical parameters
- For the thermal variants, dynamic self-heating (that is, the effect of self-heating on the electrical characteristics of the device can be simulated)

Both blocks have thermal variants, accessible under **Simscape > Block choices**. By default, the thermal port is not exposed.

Generic Battery Block Thermal and Measurement Ports: Model heat transfer from battery and access internal charge measurement

The Generic Battery block now has four modeling variants, accessible by right-clicking the block in your block diagram and then selecting 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 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 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.

When using the thermal variant, use caution when operating at temperatures outside of the temperature range bounded by the **Measurement temperature** and **Second temperature measurement** values. The block uses linear interpolation for the derived equation coefficients, and simulation results might become nonphysical outside of the specified range. The block checks that the internal series resistance, self-discharge resistance, and nominal voltage always remain positive, and issues error messages in case of violation.

Faults and Tabulated Efficiencies for Servomotor Block: Inject faults into servomotor and use tabulated efficiencies

The Servomotor block now lets you specify electrical losses by using tabulated efficiency data, instead of a single efficiency measurement or tabulated loss data.

The best practice is still to provide tabulated loss data, as a function of speed and torque, because:

- Efficiency becomes ill-defined for zero speed or zero torque.
- Using losses, you can also account for fixed losses that are still present for zero speed or torque.

If you use the tabulated efficiencies option:

- The block converts the efficiency values you provide into losses and uses the tabulated losses for simulation.

-
- Efficiency values you provide for zero speed or zero torque are ignored, and losses are assumed zero when either torque or speed is zero.
 - The block uses linear interpolation to determine losses. Provide tabulated data for low speeds and low torques, as required, to get the desired level of accuracy for lower power conditions.
 - The block does no extrapolation of loss values for speed and torque magnitudes that exceed the range of the table.

The Servomotor block now also provides variants with an exposed physical signal input port F and an additional tab, **Faults**, in the block dialog box, to let you model faults and simulate disconnecting the DC supply. You cannot simulate disconnecting the DC supply by simply opening a switch, because there must be a finite voltage on the servomotor terminals, producing the current that balances the electrical and mechanical power. See the block reference page for details.

FEM-Parameterized PMSM Block Enhancements: Use 4D flux data, Park transformation for 3D data, and improved iron losses model

The FEM-Parameterized PMSM block now lets you select between the following parameterization methods for flux and torque:

- 2-D table lookup, based on current and rotor angle. This model assumes that the mutual inductance between stator windings is independent of current and rotor angle. This is the default model, equivalent to the FEM-Parameterized PMSM block available in previous releases.
- 3-D table lookup, based on direct current, quadrature current, and rotor angle. The block uses Park transformation to map the three stator winding currents to direct and quadrature currents. This method reduces the data complexity, as compared to the 4-D table lookup, and therefore results in improved simulation performance.
- 4-D table lookup, based on the three stator winding currents and the rotor angle.

To access these parameterization methods, right-click the block in your model, select **Simscape > Block choices**, and then select the desired block variant, with or without thermal port. By default, the thermal port is not exposed.

Regardless of the parameterization methods for flux and torque, all block variants offer an improved iron losses model. Iron losses are divided into two terms, one representing the main magnetizing path, and the other representing the cross-tooth tip path that becomes active during field weakened operation.

Compatibility Considerations

The **Magnetizing resistance, R_m** parameter has been replaced by a group of parameters for the new iron losses model. You now provide two row vectors, each of length 3, of loss values, as well as the frequency at which they were measured and the resulting RMS current for the short-circuit loss test case.

The previous parameterization, using the magnetizing resistance, let you model iron losses depending on the main flux path only, and was not suitable for assessment of field weakened operation. To replicate results from the previous release, for the **Open-circuit iron losses, [P_hysteresis P_eddy P_excess]** parameter, provide a value for P_eddy only, leaving P_hysteresis and P_excess as 0.

New examples

Examples introduced in this version are:

- HEV PMSM Drive Test Harness
- PMSM Iron Losses
- AM Radio Receiver
- Interactive Generation of MOSFET Characteristics
- Interactive Generation of LDMOS Characteristics
- Interactive Generation of Thermal MOSFET Characteristics
- PWM Circuit Using 555 Timer
- DC-DC LLC Converter
- Class E DC-DC Converter
- Linear LED Driver

R2016a

Version: 2.9

New Features

Compatibility Considerations

SimElectronics renamed to Simscape Electronics

The product name SimElectronics® is changed to Simscape Electronics.

The top-level block library for the product is still located under **Simscape** in the Library Browser, but is now named **Electronics**.

Capacitor Block and Inductor Block with Fault Parameterization: Test system behavior during faults

The new Capacitor block in the Passive Devices library lets you model the following effects:

- Constant or table-based capacitance
- Nominal capacitance tolerances
- Operating limits in terms of power and working voltage range
- Fault modeling at a specific time, or when the permissible voltage range is exceeded for longer than a specific time interval

The new Inductor block in the Passive Devices library lets you model the following effects:

- Nominal inductance tolerances
- Operating limits in terms of power and saturation current
- Fault modeling at a specific time, or when the maximum voltage is exceeded more than a certain number of times, or when the maximum current is exceeded for longer than a specific time interval

Thermal Resistor block now part of Simscape Foundation library

The ability of the Resistor block to model thermal effects, introduced in R2015b on page 5-2, rendered the Thermal Resistor block superfluous. This block has now been moved to the Simscape Electrical Elements library.

There is no compatibility impact on existing models. However, MathWorks recommends that, going forward, you use the Resistor block to model thermal effects, by exposing its thermal port.

Parameters change to N-Channel IGBT, N-Channel MOSFET, and P-Channel MOSFET blocks

In previous versions, the output capacitance was implemented directly as the drain-source capacitance (for MOSFETs) or collector-emitter capacitance (for IGBT). Because the output capacitance is measured with gate and source (or gate and emitter) shorted, this implementation does not take into account the reverse transfer capacitance.

To address this issue:

- New parameters, **Drain-source junction capacitance** for MOSFETs (or **Collector-emitter junction capacitance** for IGBT), let you specify the drain-source (or collector-emitter) capacitance values directly.
- For parameterization options that derive the capacitance values, the block subtracts the reverse transfer capacitance from the output capacitance value to derive the correct drain-source (or collector-emitter) capacitance.

Compatibility Considerations

If you specified nonzero output capacitance values in your legacy models, open the corresponding block dialogs and correct the values where necessary.

New examples

Examples introduced in this version are:

- Buck Converter with Faults
- MOSFET Characterization Using Y-Parameters

R2015b

Version: 2.8

New Features

Compatibility Considerations

Fault block for modeling open-circuit and short-circuit electrical faults

Fault modeling helps you understand system behavior when a fault occurs during simulation, as well as test out fault detection and isolation algorithms. You can then stop the simulation, or just warn and continue, to see subsequent behavior.

The new Fault block in the Passive Devices library represents an electrical fault as an instantaneous change in resistance. Use it to replicate both open-circuit and short-circuit fault behaviors, and thus to represent faults on other library components. The block can trigger fault events:

- At a specific time
- When a predefined voltage range or current range is exceeded
- When an external trigger signal goes high or low

Operating limits, tolerances, and fault parameterization for Resistor block

Operating limits modeling lets you detect when components exceed their rated values during simulation. Fault modeling helps you understand system behavior when a fault occurs during simulation, as well as test out fault detection and isolation algorithms. You can then stop the simulation, or just warn and continue, to see subsequent behavior.

The Resistor block now has additional fault, tolerance, and operating limits modeling options. The block lets you model the following effects:

- Nominal resistance tolerances
- Operating limits in terms of power and maximum working voltage
- Fault modeling at a specific time, or when a current limit is exceeded for longer than a specific time interval
- Thermal noise current
- Thermal effects, such as temperature-dependent resistance and operating limits, when you expose the thermal port on the block

Compatibility Considerations

The ability of the Resistor block to model thermal effects renders the Thermal Resistor block obsolete. At this time, the Thermal Resistor block remains unchanged in the

Passive Devices library to preserve compatibility of existing models. However, this block may be removed in a future release. MathWorks recommends that, going forward, you use the Resistor block to model thermal effects, by exposing its thermal port.

FEM-Parameterized PMSM block

The new FEM-Parameterized PMSM block in the Rotational Actuators library implements a model of a permanent magnet synchronous motor (PMSM) defined in terms of magnetic flux linkage. You parameterize the block by providing tabulated data of motor magnetic flux as a function of current and rotor angle.

Servomotor block with tabulated efficiency option

Servomotor block now lets you specify efficiency using table lookup, as a function of torque and speed.

The block allows both simplified and tabulated definition of electrical losses. The default, simplified, behavior is the same as in previous releases.

Alternatively, you can provide tabulated loss values as a function of motor speed and load torque. When using this option, provide data for all of the operating quadrants that your simulation will run in. If you provide partial data (for example, just for the quadrant 1 forward motoring region), then the other quadrants are assumed to repeat the same pattern of losses. This will normally be correct for the reverse motoring region, but may be an approximation for the braking/generating quadrants. The block does no extrapolation of loss values for speed and torque magnitudes that exceed the range of the table.

R2015a

Version: 2.7

New Features

Bug Fixes

elec_getPowerLossSummary function to calculate and view semiconductor switching device losses

Checking dissipated power is useful for verifying that circuit components are operating within their working envelopes. All components that constitute blocks in the Semiconductor Devices library now have an internal variable called `power_dissipated`, which represents the instantaneous power dissipated by the component. When you log simulation data, the time-value series for this variable represents the power dissipated by the component over time. You can view and plot this data using the Simscape Results Explorer.

The `elec_getPowerLossSummary` function lets you calculate average losses for a block over a period of time. Some blocks are composite components and therefore have more than one `power_dissipated` variable, depending on their member components. For example, the N-Channel IGBT block has separate `power_dissipated` logging nodes for the MOSFET, PNP bipolar transistor, collector resistor, and emitter resistor member components. In this case, the `elec_getPowerLossSummary` function sums all these losses and provides the power loss value for the whole block.

To use the `elec_getPowerLossSummary` function, you have to enable simulation data logging and run the simulation. For more information, see [Data Logging](#).

The 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 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.

For an example of using this function to determine the efficiency of a single-stage solar converter, see [Solar Power Converter](#).

Nonlinear magnetization inductance with hysteresis option for Nonlinear Inductor and Nonlinear Transformer blocks

A new parameterization option for magnetization inductance, `Magnetic Flux Density Versus Magnetic Field Strength Characteristic with Hysteresis`, is available for the following blocks:

-
- Nonlinear Inductor
 - Nonlinear Transformer

This option lets you define magnetic flux density as a function of both the magnetization of the core and the history of the magnetic field strength, based on the Jiles-Atherton model of hysteresis. See the block reference pages for details.

Schmitt Trigger block

The Schmitt Trigger block in the Logic sublibrary of the Integrated Circuits library implements a behavioral model of Schmitt trigger. The block output logic level is HIGH when the input rises above the **High level input voltage** value and does not go LOW until the input falls below the lower-valued **Low level input voltage** value. This implements a hysteresis characteristic between input and output of the block.

Current Limiter block

The Current Limiter block in the Semiconductor Devices library provides a behavioral model of a current limiter. Use it to represent current limiting as found in some power supplies and motor drives, and also to represent components that are used to limit inrush current.

Droop parameterization for DC-DC Converter block

The DC-DC Converter block now has a new parameter, **Droop parameterization**, which lets you choose between two options:

- By voltage droop with output current — Specify the absolute value of droop by using the **Output voltage droop with output current** parameter. This is the default option and the method available in previous releases.
- By percent voltage droop at rated load — Specify droop as a percentage at rated load by using the **Percent voltage droop at rated load** parameter. This is the new method. You specify a value, in percent, by which voltage drops compared to the nominal output voltage when supplying the rated load. The default is 2 percent.

The DC-DC Converter block now also has a new physical signal input port F and a new tab, **Faults**, in the block dialog box, to let you simulate the DC supply failure and converter failure. See the block reference page for details.

Thermal port available for H-Bridge block

The H-Bridge block can now have an optional thermal port. By default, the thermal port is not displayed. To expose the thermal port, right-click the H-Bridge block in your model, and then from the context menu select **Simscape > Block choices > Show thermal port**. This action displays the thermal port H on the block icon, and adds the **Temperature Dependence** and **Thermal port** tabs to the block dialog box.

When the thermal port is visible:

- The heat generated by the bridge on-resistance and freewheeling diodes is added to the thermal port. The thermal port has an associated thermal mass and initial temperature that are set from the **Thermal port** tab.
- The bridge on-resistance and freewheeling diode resistance become functions of temperature. The values for these resistances and the second measurement temperature are defined from the **Temperature Dependence** tab. Resistance is assumed to vary linearly between the two measurement temperatures. Extrapolation is used for temperatures outside of this range except for when simulating in averaged mode with discontinuous load current characteristics.

New examples

Examples introduced in this version are:

- Inductor With Hysteresis
- Analog Anti-Aliasing Filter
- Phase-Locked Loop

R2014b

Version: 2.6

New Features

Averaged mode for stepper motor and driver blocks

A new parameter, **Simulation mode**, lets you select between Stepper and Averaged modes, and is available for the following blocks:

- Stepper Motor
- Stepper Motor Driver
- Unipolar Stepper Motor
- Unipolar Stepper Motor Driver

If you select Averaged simulation mode for a motor-driver pair (that is, for a Stepper Motor block and for the Stepper Motor Driver block that controls it, or for a Unipolar Stepper Motor block and its Unipolar Stepper Motor Driver block), then the individual steps are not simulated. This can be a good way to speed up simulation.

Averaged mode includes a slip estimator to predict whether the stepper motor would have slipped if running in Stepping simulation mode. Upon detecting slip, the simulation either continues (with or without a warning) or stops with an error, depending on your setting for the **Action on slipping** parameter. If you simulate or predict slip, it is generally a good practice to do some validation runs comparing Stepping and Averaged modes before using the averaged model representation for simulation studies. See the block reference pages for details.

Torque or force calculation option for FEM actuator blocks

FEM-Parameterized Rotary Actuator and FEM-Parameterized Linear Actuator blocks let you either enter the torque or force data, as in previous releases, or calculate the torque or force matrix from flux linkage information.

If the finite element package does not provide torque (force) and provides only flux data, then you can let the block automatically calculate the torque (force) matrix from the flux information. To select this option, set the **Calculate torque matrix?** or **Calculate force matrix?** parameter (as appropriate) to **Yes**. The torque (force) matrix calculation occurs at model initialization based on current block flux linkage information. See the block reference pages for details.

Nonzero threshold width in Switch blocks

A new parameter, **Threshold width**, is available for the following blocks:

-
- SPST Switch
 - SPDT Switch
 - DPST Switch
 - DPDT Switch

If the **Threshold width** parameter is set to zero (which is the default), the switch is closed if the voltage presented at the v_T control port exceeds the value of the **Threshold** parameter.

If the **Threshold width** parameter is greater than zero, then switch conductance G varies smoothly between off-state and on-state values. Defining a small positive **Threshold width** can help solver convergence in some models, particularly if the control port signal v_T varies continuously as a function of other network variables. However, defining a nonzero threshold width precludes the solver from making use of switched linear optimizations. Therefore, if the rest of your network is switched linear, MathWorks recommends that you set **Threshold width** to zero.

Event-based variant of the N-Channel IGBT block

In previous releases, the N-Channel IGBT block provided two ways of modeling an IGBT:

- As an equivalent circuit based on a PNP bipolar transistor and N-channel MOSFET
- By a lookup table approximation to the I-V (current-voltage) curve

These two methods are still available under the **Full I-V and capacitance characteristics** block variant. In documentation, this variant is also referred to as the detailed model. A new block variant, **Simplified I-V characteristics and event-based timing**, provides a simplified, event-based model, which lets you represent the IGBT more simply by using just the on-state I-V data corresponding to the gate voltage used in your circuit. Switching between states is achieved by linearly ramping the collector-emitter voltage, which results in much faster simulation speed. Use the event-based variant when the focus of the analysis is to understand overall circuit behavior rather than to verify the precise IGBT timing or losses characteristics. For details, see Event-Based IGBT Variant.

To select the desired variant, right-click the N-Channel IGBT block in your model. From the context menu, select **Simscape > Block choices**, and then one of the following options:

- **Full I-V and capacitance characteristics | No thermal port** — Detailed model that does not simulate the effects of generated heat and device temperature. This is the default.
- **Full I-V and capacitance characteristics | Show thermal port** — Detailed model with exposed thermal port.
- **Simplified I-V characteristics and event-based timing | No thermal port** — Simplified event-based model, which also does not simulate the effects of generated heat and device temperature.
- **Simplified I-V characteristics and event-based timing | Show thermal port** — Simplified event-based model with exposed thermal port.

See the N-Channel IGBT block reference page for more information.

Tabulated capacitance options for MOSFET and IGBT blocks

N-Channel IGBT, N-Channel MOSFET, and P-Channel MOSFET blocks now have additional options on the **Junction Capacitance** tab that let you parameterize the charge model using table lookup.

In previous releases, these blocks provided two options for modeling capacitance: either by specifying the junction capacitance values directly, or by letting the block derive them from the input and reverse transfer capacitance values. Now each of these methods allows you to enter either fixed capacitance values or tabulated values as a function of the collector-emitter (drain-source) voltages. See the block reference pages for details.

Z pulse width change in Incremental Shaft Encoder block

In previous releases, there was no correlation between the Z pulse width and the length of the A and B pulses. Therefore, although the block created a rising edge for the synchronization Z pulse at the correct time, you could not use this block to synchronize on the falling edge.

The Z pulse on time is now equal to the A and B pulse on times for a given rotational speed, giving you the ability to decode on the rising edge and on the falling edge. For existing models where just the rising edge of the pulse trains is used for decoding or where the Z pulse is not used, there is no change in behavior.

New examples

Examples introduced in this version are:

- IGBT Behavioral Model
- Unipolar Stepper Motor Averaged Mode

R2014a

Version: 2.5

New Features

Compatibility Considerations

Unipolar Stepper Motor Driver block and half-stepping option for stepper motor drivers

The new Unipolar Stepper Motor Driver block in the Drivers library represents a driver specifically configured for use with the Unipolar Stepper Motor block. It connects the two winding center-tap connections A0 and B0 to the positive supply with a voltage equal to the value you provide for the **Output voltage amplitude** parameter. The A+, A-, B+ and B- ports are then grounded in the appropriate sequence to create the stepping motion. The block initiates a step each time the voltage at the PWM port rises above the **Enable threshold voltage**.

The Stepper Motor Driver block now has an option to produce the output waveforms required for half-stepping. By default, the driver operates in full-stepping mode, but if you set the **Stepping mode** parameter to `Half stepping`, the driver generates an intermediate state between the full steps, in which just one of the A or the B half-windings is powered. As a result, the step size is half of the stepper motor's full step size. The new Unipolar Stepper Motor Driver block also has this option.

Detent torque for stepper motors

Both the Stepper Motor and the Unipolar Stepper Motor blocks now let you account for the torque variation observed when the motor is unpowered and the shaft is rotated. Detent torque is generated by the interaction of the permanent magnet flux and the variation in reluctance as the teeth come in and out of alignment. The new **Detent torque** parameter represents the amplitude of the sinusoidal torque variation, and the corresponding term is added to the block equations. By default, the parameter value is 0, in which case detent torque is neglected in block computations. Legacy models are not affected.

Output capacitance for MOSFET and IGBT blocks

A new block parameter lets you specify a value for output capacitance:

- For N-Channel IGBT, the **Output capacitance, Coes** parameter represents the output capacitance applied across the collector-emitter ports.
- For N-Channel MOSFET and P-Channel MOSFET, the **Output capacitance, Coss** parameter represents the output capacitance applied across the drain-source ports.

Modeling the output capacitance is generally important in signal processing applications, as it has measurable effect on circuit behavior. For motor control applications the effect

is not as significant, because the other capacitances play a much bigger part in setting turn-on and turn-off dynamics. However, if your datasheet provides a value for output capacitance, you can enter it to increase the block fidelity. The default value is 0, therefore legacy models are not affected.

Lookup table definition of I-V relationship in IGBT and Thyristor blocks

In previous releases, the Thyristor and N-Channel IGBT blocks have been represented by an equivalent circuit constructed from other semiconductor blocks, such as bipolar transistors and MOSFETs. Now both blocks have an additional representation option, by a lookup table approximation of the on-state current-voltage curve. The main advantages of using this option are increased simulation speed and ease of parameterization. See the block reference pages for details.

Thyristor block that now models devices with higher leakage currents

In the Thyristor block represented by an equivalent circuit, the underlying model has been enhanced with a pair of back-to-back diodes to allow a wider range of leakage current variation. To help you parameterize your thyristor to match the manufacturer datasheet, two test harnesses have been added to the SimElectronics examples, Thyristor Static Behavior Validation and Thyristor Dynamic Behavior Validation. See the block reference page for more information.

Compatibility Considerations

To take advantage of the additional diodes, the mapping of the block parameters to the equation parameters has been changed. Existing models may need fine-tuning of parameter values to match the exact behavior with previous versions of the block, particularly for parameters associated with dynamics.

New examples

Examples introduced in this version are:

- Switched Reluctance Motor Parameterized with FEM data
- Automotive Electrical System
- Photovoltaic Generator

- Microcontroller with GPIO, ADC and DAC Connections
- IGBT Loss Characteristics

R2013b

Version: 2.4

New Features

Voltage-Controlled Oscillator block

The new Voltage-Controlled Oscillator block in the Integrated Circuits library models abstracted behavior of a voltage-controlled oscillator (VCO). VCOs are basic building blocks for RF transmitter and receiver devices, microprocessor clocks, and A-to-D converters. The block models a behavioral representation of a VCO for simulation efficiency.

You have two options for specifying frequency dependence on the input voltage:

- *Linear* — By specifying a coefficient for the rate of change of frequency with input voltage.
- *Tabulated* — By using a lookup table and specifying a vector of input voltages and a corresponding vector of VCO frequencies relative to the nominal frequency.

You also can model the time delay between a change in the input control voltage and the oscillator frequency.

Servomotor block that now permits parameterized torque-speed curve and optional iron losses

In previous versions, the Servomotor block allowed you to specify the torque-speed envelope only as a set of speed data points and corresponding maximum torque values. Now, you can also define the torque-speed envelope by specifying a maximum torque and a maximum power instead of providing the tabulated torque-speed data.

The block now also accounts for iron losses due to eddy currents. The **Torque-independent electrical losses** parameter is renamed to **Fixed losses independent of torque and speed**, and the new **Iron losses** parameter lets you specify the iron losses at the speed and torque at which efficiency is defined.

Solar Cell block that now models solar cell arrays

In previous versions, the Solar Cell block allowed you to model a single photovoltaic cell. Now, you can model any number of solar cells connected in series using a single Solar Cell block by setting the parameter **Number of series cells** to a value larger than one. Internally the block still simulates only the equations for a single solar cell, but scales up the output voltage according to the number of cells. This results in a more efficient simulation than if equations for each cell were simulated individually.

If you want to model N cells in parallel, you can do so for single cells by scaling the parameter values accordingly. That is, multiply short-circuit current, diode saturation current, and solar-generated currents by N , and divide series resistance by N . To connect solar cell blocks in parallel, where each block contains multiple cells in series, make multiple copies of the block and connect accordingly.

New examples

Examples introduced in this version are:

- Solar Cell Power Curve
- Solar Power Converter

R2013a

Version: 2.3

New Features

Compatibility Considerations

Resistor with optional thermal noise

The new Resistor block in the Passive Devices library models a linear resistor that can optionally generate thermal noise current. The Gaussian noise is generated using the new Random Number source block in the Simscape Foundation library. The **Repeatability** parameter gives you the following options for noise control: not repeatable, repeatable with a random seed, and repeatable with explicitly specified seed.

Compatibility Considerations

Legacy models, created in R2009b or earlier and not saved using a more recent version, may need updating if they contain the SPICE Resistor block. If this is the case, upon opening the model you will get a warning saying that the Resistor block does not have a parameter named SCALE. To fix this issue, replace the resistor either with the new Resistor block from the Passive Devices library, or with a new copy of the SPICE Resistor block from the Additional Components/Spice-Compatible Components library.

Voltage and current sources with DC, AC, and noise components

Two new blocks have been added to the Sources library:

- Voltage Source block — Configurable as either AC or DC voltage source with optional noise
- Current Source block — Configurable as either AC or DC current source with optional noise

By default, both AC and DC components are set to 0 in each block. Define the AC/DC voltage or current by specifying nonzero values for the appropriate parameters after placing the block in your model.

If you enable the noise component, the Gaussian noise voltage or current, respectively, is generated using the new Random Number source block in the Simscape Foundation library. The **Repeatability** parameter gives you the following options for noise control: not repeatable, repeatable with a random seed, and repeatable with explicitly specified seed.

Nonlinear charge model for N-Channel MOSFET, P-Channel MOSFET, and N-Channel IGBT blocks

The two MOSFET blocks now model gate junction capacitance as a fixed gate-source capacitance and either a fixed or a nonlinear gate-drain capacitance. The IGBT block now models gate junction capacitance as a fixed gate-emitter capacitance and either a fixed or a nonlinear gate-collector capacitance.

If you select the nonlinear option for the charge-voltage linearity, then the gate-drain charge relationship, or the gate-collector charge relationship, respectively, is defined by a piecewise-linear function. For more information, see the respective block reference pages.

New examples

Examples introduced in this version are:

- Op-Amp with Noise
- Delta Sigma ADC with Noise
- IGBT Dynamic Characteristics

R2012b

Version: 2.2

New Features

Compatibility Considerations

Single-pole and double-pole switch blocks

The Passive Devices library now contains a sublibrary, named Switches, with four new blocks:

- The DPDT Switch block models a double-pole double-throw switch.
- The DPST Switch block models a double-pole single-throw switch.
- The SPDT Switch block models a single-pole double-throw switch.
- The SPST Switch block models a single-pole single-throw switch.

For each switch type, you have an option of modeling turn-on and turn-off delays, that is, a delay between the point at which the voltage at the control port passes the threshold and the point at which the switch opens or closes.

Five new sensor blocks: PTC Thermistor, Pressure Transducer, Resolver, Gyro, and Accelerometer

Five new blocks have been added to the Sensors library:

- The Accelerometer block provides an abstract model of a MEMS accelerometer. The acceleration at the mechanical translational port is mapped to either a voltage level or the duty cycle of a PWM voltage across the electrical + and - ports. The output voltage is limited according to the values that you provide for maximum and minimum output voltage. Optionally, you can model sensor dynamics by adding a first-order lag between the angular rate at the mechanical port and the corresponding voltage applied to the electrical + and - ports.
- The Gyro block provides an abstract model of a MEMS gyroscope. The gyro provides an output voltage that is proportional to the angular rotation rate presented at the mechanical rotational port. The output voltage is limited according to the values that you provide for maximum and minimum output voltage. Optionally, you can model sensor dynamics by adding a first-order lag between the angular rate at the mechanical port and the corresponding voltage applied to the electrical + and - ports.
- The PTC Thermistor block models a switching type positive temperature coefficient (PTC) thermistor. This type of thermistor has a decreasing resistance with increasing temperature, up to the Curie temperature. Above the Curie temperature, the resistance increases very rapidly with increasing temperature. To represent a non-switching linear PTC thermistor, use the Thermal Resistor block.
- The Pressure Transducer block models a generic pressure transducer that turns a pressure measurement into a voltage. The output voltage is linearly proportional to

the pressure. If the pressure is less than zero, the block outputs zero volts. An input pressure equal to the **Pressure range** parameter value results in an output voltage equal to the **Full-scale deflection** parameter value. For higher pressures, the output voltage remains at this **Full-scale deflection** value. You have three choices of operation mode, which let you select between vacuum, atmospheric pressure, or sealed-gauge reference pressure as the reference point for the pressure measurement. You also have an option of approximating the sensor dynamics by a first-order lag.

- The Resolver block models a generic resolver, which consists of a rotary transformer that couples an AC voltage applied to the primary winding to two secondary windings. These secondary windings are physically oriented at 90 degrees to each other. As the rotor angle changes, the relative coupling between the primary and the two secondary windings varies. In the Resolver block model, the first secondary winding is oriented such that peak coupling occurs when the rotor is at zero degrees, and therefore the second secondary winding has minimum coupling when the rotor is at zero degrees.

Nonlinear Transformer and Nonlinear Inductor blocks

Two new blocks in the Passive Devices library let you take into account 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. You have multiple options of block parameterization, including single inductance, single saturation point, magnetic flux versus current characteristic, and magnetic field density versus magnetic field strength characteristic.
- 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 primary and secondary values, or by separate values for primary and secondary leakage resistance and inductance.

DC-DC Converter block and Incandescent Lamp block

The new DC-DC Converter block in the Sources library represents a behavioral model of a power converter. The power converter regulates voltage on the load side and the required amount of power is drawn from the supply side to balance input power, output power, and losses. Optionally, the converter can support regenerative power flow from load to supply.

The new Incandescent Lamp block in the Passive Devices library models an incandescent lamp. The key characteristic of this block is that the resistance increases as the filament warms up. The rate of heat loss from the filament is proportional to the filament's temperature difference to ambient. Optionally, you can simulate the fault dynamics by specifying a simulation time at which the lamp fails.

Unipolar Stepper Motor block and Stepper Motor block that now account for iron losses

The new Unipolar Stepper Motor block in the Rotational Actuators library represents a stepper motor that has center taps on the two phase windings. All four half-windings are identical. The block lets you simulate thermal effects, and also accounts for iron losses, both in its electrical and thermal equations.

The existing Stepper Motor block has been enhanced to account for iron losses, as well. It now has an additional thermal port HR, corresponding to the rotor and associated iron losses. Additional parameters on the **Electrical** and **Thermal Port** tabs let you specify magnetizing resistance, rotor thermal mass and initial temperature, and the percentage of the magnetizing resistance associated with the magnetic path through the rotor.

Compatibility Considerations

If your existing model uses a Stepper Motor block with thermal ports exposed, the block now has an additional thermal port HR (corresponding to the rotor and associated iron losses). Leaving this new port unconnected results in a simulation-time error. If iron losses associated with the rotor are not important, you can connect the HR port to a Perfect Insulator block.

New examples

Examples introduced in this version are:

- Automotive Alternator Charging a Battery
- Nonlinear Inductor Characteristics
- Nonlinear Transformer Characteristics
- TVS Diode Parameterization
- Ultracapacitor With Converter

-
- Power Split Hybrid Vehicle Electrical Network

R2012a

Version: 2.1

New Features

Thermal Ports Available for Actuator Blocks and for Solar Cell Block

All the blocks in the Rotational Actuators and Translational Actuators libraries, as well as the Solar Cell block in the Sources library, can now have optional thermal ports. By default, the thermal ports are not displayed. To expose the thermal port, right-click on the relevant block in your model, and from the context menu select **Simscape block choices > Show thermal port**. This action displays the thermal port H on the block icon, and adds the **Thermal port** tab to the block dialog box.

This functionality is not always available for blocks in existing models, depending on when the model was last saved. If you right-click on a block in a model saved in a previous version, and the context menu item **Simscape block choices** does not appear, make a new copy of the block from the SimElectronics library.

Fully Differential Op-Amp Block

The new Fully Differential Op-Amp block in the Integrated Circuits library models an operational amplifier with fully differential output, that is, not referenced to ground. The output common-mode voltage is controlled by the common-mode port *cm*. Internal resistors set the nominal output common-mode voltage to be midway between the values you provide for the positive and negative supply voltages. Applications include data acquisition where inputs are differential, for example, sigma-delta converters.

The block provides a behavioral model of a fully differential operational amplifier. It does not represent nonlinear effects, such as variation in gain with output voltage amplitude, and the nonlinear nature of the output voltage-current relationship for large load currents.

Transmission Line Block

The new Transmission Line block in the Passive Devices library lets you model a transmission line either by using delays, or by a lumped parameter model. Use the delay-based models for better simulation performance at system level. The lossless delay-based model represents an ideal transmission line.

Power Sensor Block

The new Power Sensor block in the Sensors library calculates the power taken by the load connected across the + and - terminals under the assumption that only the load is connected to the + terminal.

The sensor can return either instantaneous power, or power averaged over a fixed time period. The latter option is useful when working with periodic current and voltage waveforms, such as those associated with PWM control.

For an example of using this block, see the Flyback Converter demo.

Induction Motor Block Usability Enhancement

The usability of the Induction Motor block has been improved. In previous versions, when setting **Model parameterization** to `By motor ratings`, you had to provide a value for either the motor starting current or maximum torque. This group of parameters has been removed, and instead the **Rated RMS line current** parameter value is used to determine the total motor inductance. In existing models, if you used consistent values for **RMS starting (or locked rotor) line current** and **Rated RMS line current**, simulation results are the same as in previous versions.

Controlled PWM Voltage Block Enhancement

The Controlled PWM Voltage block has two new parameters, **Pulse delay time** and **Pulse width offset**. Use these parameters to add a small turn-on delay and a small turn-off advance. This can be helpful when fine-tuning switching times, to minimize switching losses.

Exponential Diode Block Now Models Charge Dynamics

The Diode block now lets you model charge dynamics. The Exponential diode model contains an additional set of parameters that let you either specify values for the transit time and carrier lifetime directly, or calculate them using the peak reverse current and reverse recovery time. This functionality is especially useful for applications such as commutation diodes.

New Demos

Demos introduced in this version are:

- Controllable Phase Shifter
- Low-Pass Filter Using Operational Transconductance Amplifiers
- Fourth-Order Sallen-Key Lowpass Filter

- Flyback Converter

R2011b

Version: 2.0

New Features

Compatibility Considerations

Thermal Ports Available for Semiconductor Blocks

All the blocks in the Semiconductors library, as well as the Photodiode and Light-Emitting Diode blocks in the Sensors library, can now have optional thermal ports. By default, the thermal ports are not displayed. To expose the thermal port, right-click on the relevant block in your model, and from the context menu select **Simscape block choices > Show thermal port**. This action displays the thermal port H on the block icon, and adds the **Thermal port** tab to the block dialog box.

Compatibility Considerations

In the N-Channel IGBT block, several new parameters on the **Advanced** tab have been added, to better match the typical device datasheets. The **Forward Early voltage, VAF** parameter, with the default value of 200 V, specifies the Forward Early voltage for the PNP transistor. Previously the effect was not modeled. This means that existing models will show small differences in the current-voltage relationship associated with the PNP bipolar transistor, compared to the previous version. Additionally, the new **Collector resistance, RC** and **Emitter resistance, RE** parameters have nonzero default values, to improve the numeric efficiency of computations. If you want to preserve the simulation results for the existing models, set **Forward Early voltage, VAF** to $1e10$, **Collector resistance, RC** to 0, and **Emitter resistance, RE** to 0.

New Operational Transconductance Amplifier Block

The Operational Transconductance Amplifier block, added to the Integrated Circuits library, provides a behavioral representation of an operational transconductance amplifier. A transconductance amplifier converts an input voltage into an output current. Applications include variable frequency oscillators, variable gain amplifiers, and current-controlled filters. These applications are based on the fact that the transconductance gain is a function of current flowing into the control current pin.

The block does not model the detailed transistor implementation. This results in faster simulation, but the model is only valid when operating in the linear region, that is, where the device input resistance, output resistance, and transconductance gain all depend linearly on the control current, and are independent of input signal amplitude.

New Push-Pull Output Block

The Push-Pull Output block, added to the Integrated Circuits library, provides a behavioral representation of a CMOS complementary output stage. To improve

simulation speed, the block does not model all the internal individual MOSFET devices that make up the gate. You can use this block to create a representative output current-voltage relationship when defining an integrated circuit model behavior with Physical Signal blocks from the Simscape Foundation library. For an example, see the Modeling an Integrated Circuit demo.

H-Bridge Block Enhancements

The following enhancements have been implemented for the H-Bridge block:

- In Averaged mode, a new **Load current characteristics** parameter is available with two options, `Smoothed` and `Unsmoothed` or `discontinuous`. The first option assumes that the current is practically continuous due to load inductance, and corresponds to the old block behavior. For cases where the current is not smooth, or goes to zero between PWM cycles, use the `Unsmoothed` or `discontinuous` option, and provide values for the new **Total load series resistance**, **Total load series inductance**, and **PWM frequency** parameters. During simulation, the block uses these values to calculate a more accurate value for H-bridge output voltage that achieves the same average current as would be present if simulating in PWM mode.
- The **Freewheeling mode** parameter is now available not only in PWM mode, but also in Averaged mode in cases where you select `Unsmoothed` or `discontinuous` for the **Load current characteristics** parameter.
- An additional **Freewheeling mode** option, `Via two semiconductor switches and one freewheeling diode`, controls the load by maintaining one high-side bridge arm permanently on and using the PWM signal to toggle between enabling the corresponding low-side bridge arm and the opposite high-side bridge arm. This means that the block uses a freewheeling diode in parallel with a bridge arm, plus another series bridge arm, to complete the dissipation circuit when the bridge turns off.
- The block dialog box has been reorganized using tabs, to improve usability.

DC Motor Block Supports No-Load Current Data for Rotor Damping

The DC Motor block has an additional option that lets you use no-load current data to calculate a value for rotor damping. This is helpful when the manufacturer datasheet does not provide an explicit rotor damping value.

The **Rotor damping parameterization** drop-down has been added to the **Electrical Torque** tab of the block dialog box, with the following values:

- By damping value — Specify a value for rotor damping directly, by using the **Rotor damping** parameter on the **Mechanical** tab. This is the default.
- By no-load current — The block calculates rotor damping based on the values that you specify for the **No-load current** and **DC supply voltage when measuring no-load current** parameters. If you select this option, the **Rotor damping** parameter is not available on the **Mechanical** tab.

Compatibility Considerations

Previously, if the **Model parameterization** parameter was set to *By stall torque & no-load speed* or *By rated power, rated speed & no-load speed*, the block did not take rotor damping into account. The new block equations always include rotor damping, because it is now tied to no-load current. Therefore, rated speed and no-load speed results for existing models using these options will be slightly different than in previous versions if the model has a nonzero damping value.

If you wish to retain the original behavior, set the rotor damping to zero, and add an external Rotational Damper block (from Simscape Foundation library) across the motor R and C ports.

New Demos

Demos introduced in this version are:

- IGBT Thermal Characteristics
- Low-Cost Voltage Regulator
- Synchronous Buck Converter With Thermal Dynamics
- Thermal Characteristics of a Synchronous Buck Converter

R2011a

Version: 1.6

New Features

Compatibility Considerations

Thermal Dependency Added to Semiconductor Blocks

Dialog boxes of most of the blocks in the Semiconductors library, and some related blocks, now have a new tab, **Temperature Dependence**, which lets you specify additional parameters to model the temperature dependence during simulation. For details, see reference pages of the following blocks:

- Diode
- Light-Emitting Diode
- N-Channel IGBT
- N-Channel JFET
- N-Channel MOSFET
- NPN Bipolar Transistor
- Optocoupler
- Photodiode
- P-Channel JFET
- P-Channel MOSFET
- PNP Bipolar Transistor

Compatibility Considerations

In NPN and PNP Bipolar Transistor blocks, a new parameter, **Collector-emitter voltage at which h-parameters are defined**, has been added. It serves to increase the accuracy with which equation parameters are calculated from h-parameters, to better capture current gain dependence on temperature. As a result, when you use `Specify` from a datasheet for the **Parameterization** parameter, there is a small change in the resulting transistor gain BF (calculated from the **Forward current transfer ratio h_{fe}** parameter value), compared to the previous version of the block.

New Demos

Demos introduced in this version are:

- Torque Motor
- Schottky Barrier Diode Characteristics

-
- IGBT Characteristics
 - Master-Slave J-K Flip-Flop

Change to an existing demo:

- The Finite Element Parameterized Solenoid demo now includes comparison with the Simscape solenoid demo `ssc_solenoid.mdl`, to illustrate the effects of flux saturation.

R2010b

Version: 1.5

New Features

New Thyristor Block

The new Thyristor block, located in the Semiconductor Devices library, represents a thyristor modeled using an NPN and a PNP transistor. The collector of each device is connected to the base of the other device so as to give the P-N-P-N junction structure of a thyristor.

New Multiplier Block

The new Multiplier block, located in the Integrated Circuits library, represents an integrated circuit multiplier for physical signals. It allows you to multiply and divide signals without switching to ulink signals and back.

Additional Exponential Diode Parameterization Options

When using the Diode block, with the **Diode model** parameter set to **Exponential**, you now have two additional options under **Parameterization**:

- Use an I-V data point and **IS** — Specify measured data at a single point on the diode I-V curve in combination with the saturation current.
- Use an I-V data point and **N** — Specify measured data at a single point on the diode I-V curve in combination with the emission coefficient.

See the block reference page for details.

Channel Modulation Parameter Added for MOSFET Blocks

The N-Channel MOSFET and P-Channel MOSFET blocks now have an additional parameter, **Channel modulation, L**. The default value is 0 1/V. See the respective block reference pages for details.

Changes to the Bipolar Transistor Blocks

The following changes have been implemented in the NPN Bipolar Transistor and PNP Bipolar Transistor blocks:

- The **Junction Capacitance** tab has been renamed to **Capacitance**, and the two existing parameters on it have been renamed:

-
- **Base-emitter capacitance to Base-emitter junction capacitance**
 - **Base-collector capacitance to Base-collector junction capacitance**
 - Two new parameters have been added to the **Capacitance** tab:
 - **Total forward transit time**, representing the mean time for the minority carriers to cross the base region from the emitter to the collector
 - **Total reverse transit time**, representing the mean time for the minority carriers to cross the base region from the collector to the emitter
 - Default values for ohmic resistances have been changed to $R_B = 1 \Omega$, $R_C = 0.01 \Omega$, and $R_E = 1e-4 \Omega$, to be consistent with the SPICE-compatible library.

New Demos

Demos introduced in this version are:

- Thyristor Static Behavior Validation
- Thyristor Dynamic Behavior Validation
- IC Multiplier Circuits
- Synchronous Buck Converter

R2010a

Version: 1.4

New Features

Compatibility Considerations

New Linear and Rotary Motors Defined in Terms of Flux

Two new blocks represent models of a motor or actuator defined in terms of magnetic flux:

- FEM-Parameterized Linear Actuator block, located in the Translational Actuators library
- FEM-Parameterized Rotary Actuator block, located in the Rotational Actuators library

New Potentiometer Block

The new Potentiometer block, located in the Passive Devices library, represents a potentiometer, where the wiper position is controlled by the input physical signal.

Initial Conditions Tab Added for Logic Blocks

The dialog boxes of blocks in the Logic library now have an additional tab, Initial Conditions, which lets you specify the output initial state (low or high). See the respective block reference pages for details.

Changes in Block Parameterization

The ability to parameterize SimElectronics blocks by importing circuit data from a SPICE netlist is no longer supported. As a result, using the `netlist2sl` function is no longer recommended. See *Parameterizing Blocks from Datasheets* for alternative ways of block parameterization. Additional related changes introduced in this version are:

- “Changes to the SPICE-Compatible Blocks” on page 16-2
- “Changes to the Solar Cell Block” on page 16-3

Changes to the SPICE-Compatible Blocks

The SPICE-compatible blocks have been moved to the Additional Components library. They are organized in sublibraries according to function, for example, the SPICE-Compatible Sources library is now the Sources sublibrary of the Additional Components/SPICE-Compatible Components library. The Resistor block, renamed SPICE Resistor, and the Current-Controlled Switch and Voltage-Controlled Switch blocks have been

moved to the Passive Devices sublibrary of the Additional Components/SPICE-Compatible Components library.

Some of the blocks have been renamed so that their names start with the “SPICE” prefix. The following table lists the old and new block names.

Old Name	New Name
Diode (SPICE)	SPICE Diode
NJFET	SPICE NJFET
NMOS	SPICE NMOS
NPN	SPICE NPN
PJFET	SPICE PJFET
PMOS	SPICE PMOS
PNP	SPICE PNP
Resistor	SPICE Resistor

There are no compatibility considerations as a result of renaming the SPICE-compatible blocks and moving them to the Additional Components library. Your existing models will be updated automatically when you open and save them in the new version.

Changes to the Solar Cell Block

In previous versions, the Solar Cell block had the option of using the SPICE Environment Parameters block to set temperature. This is removed in R2010a to eliminate dependency on the SPICE sublibrary. Also, the Solar Cell model now uses the regular Diode block (exponential diode) rather than the SPICE Diode block.

Compatibility Considerations

There is an insignificant change in results, of the order of $1e-12$, in the Solar Cell block because of the diode replacement.

New Demos

Demos introduced in this version are:

- Finite Element Parameterized Solenoid

- Circuit Level Switched Capacitor ADC
- Switching Audio Power Amplifier
- Bridge Configuration Switching Audio Power Amplifier
- Differential Pair Amplifier
- Low-Noise Bipolar Transistor Voltage Amplifier
- Triangle Wave Generator
- LC Transistor Oscillator
- Voltage-Controlled Oscillator with PI Control
- Voltage Regulator
- Band-Pass Filter Using Three Mutually-Coupled Inductors
- Class-E RF Amplifier
- Diode Ring Demodulator
- LC Transmission Line and Test Bridge

Functions and Function Elements Being Removed

Function or Function Element Name	What Happens When you use the Function or Element?	Use This Instead	Compatibility Considerations
netlist2sl	Issues a warning that it is not supported and may be removed in future releases	See Parameterizing Blocks from Datasheets for alternative ways of block parameterization	See “Changes in Block Parameterization” on page 16-2

R2009b

Version: 1.3

New Features

Compatibility Considerations

Actuators & Drivers Library Blocks

New features and changes introduced in this version are:

- “New Generic Rotary Actuator Block” on page 17-2
- “New Generic Linear Actuator Block” on page 17-2
- “Improved Servomotor Block” on page 17-2

New Generic Rotary Actuator Block

The Generic Rotary Actuator block models the torque-speed characteristics of a generalized rotary actuator.

New Generic Linear Actuator Block

The Generic Linear Actuator block models the force-speed characteristics of a generalized linear actuator.

Improved Servomotor Block

The Servomotor block now allows for the specification of additional parameters from within the Block Parameters dialog box.

Compatibility Considerations

During simulation, the updated Servomotor block is backwards-compatible with models defined in earlier versions of the software. However, the model generates a warning in this version because the block dialog box supports additional unit options for torque and speed data. To remove the warnings, open the block dialog box and select appropriate units for the torque and speed data.

New Abstracted Timer Block

The new Timer block, located in the Integrated Circuits library, is an abstracted behavioral model of a timer integrated circuit, such as the NE555.

New Demos

Demos introduced in this version are:

-
- Brushless DC Motor
 - ARINC 429 Communications Link
 - PNP Bipolar Transistor Characteristics

R2009a

Version: 1.2

New Features

Actuators & Drivers Library

New features and changes introduced in this version are:

- “New Piezo Motor Blocks” on page 18-2
- “Enhanced H-Bridge Block” on page 18-2

New Piezo Motor Blocks

The Actuators & Drivers library now contains blocks for modeling piezoelectric travelling wave motors. The library contains these new blocks:

- The Piezo Rotary Motor models the torque-speed characteristics of a rotary piezoelectric motor.
- The Piezo Linear Motor models the force-speed characteristics of a linear piezoelectric motor.

Enhanced H-Bridge Block

The H-Bridge block now provides the option to dissipate current via two freewheeling diodes when the signal at the PWM port is low. To use this new option, select `Via two freewheeling diodes` for the `Freewheeling mode` parameter.

Passive Devices Library

New features and changes introduced in this version are:

- “New Switch Blocks” on page 18-2
- “New Resistor Block” on page 18-3
- “New Crystal Block” on page 18-3
- “Enhanced Variable Inductor and Variable Capacitor Blocks” on page 18-3

New Switch Blocks

The Passive Devices library now contains Current-Controlled Switch and Voltage-Controlled Switch blocks to model electrical switches with hysteresis.

New Resistor Block

The Passive Devices library now contains a Resistor on page 16-2 block to model a resistor as a function of temperature and process data.

New Crystal Block

The Passive Devices library now contains a Crystal block to model the electrical characteristics of a crystal resonator.

Enhanced Variable Inductor and Variable Capacitor Blocks

The Variable Inductor and Variable Capacitor blocks have the following enhancements:

- The Variable Inductor block now provides two options for the relationship between the voltage across the device and the current through the inductor. The new **Equation** parameter lets you select the voltage-current equation that you want.
- The Variable Capacitor block now provides two options for the relationship between the current through the device and the voltage across the capacitor. The new **Equation** parameter lets you select the current-voltage equation that you want.

Sources Library

New features and changes introduced in this version are:

- “Enhanced Solar Cell Block” on page 18-3
- “New Two-Input Dependent Source Blocks” on page 18-3

Enhanced Solar Cell Block

The Solar Cell block has the following enhancements:

- The block now provides the option to use an 8-parameter model that includes an additional diode and a parallel resistor.
- The block now models temperature dependence.

New Two-Input Dependent Source Blocks

The SPICE-Compatible Sources library (in the Sources library) contains blocks for modeling dependent sources with two controlling inputs. The library contains these new blocks:

- PCCCS2 — Model polynomial current-controlled current source with two controlling inputs
- PCCVS2 — Model polynomial current-controlled voltage source with two controlling inputs
- PVCCS2 — Model polynomial voltage-controlled current source with two controlling inputs
- PVCVS2 — Model polynomial voltage-controlled voltage source with two controlling inputs

SPICE-Compatible Semiconductors Library

New features and changes introduced in this version are:

Enhanced NMOS and PMOS Blocks

The NMOS and PMOS blocks now provide the option to model the electrical characteristics of SPICE Level-3 MOSFET devices.

R2008b

Version: 1.1

New Features

New CMOS Logic Gate Blocks

The Logic library (in the Integrated Circuits library) contains blocks for modeling CMOS logic gates behaviorally. The library contains these new blocks:

- CMOS AND
- CMOS Buffer
- CMOS NAND
- CMOS NOR
- CMOS NOT
- CMOS OR
- CMOS XOR

New Piezo Stack Block

The Actuators & Drivers library now contains a Piezo Stack block to model the electrical and force characteristics of a piezoelectric stacked actuator.

New Relay Block

The Passive Devices library now contains a Relay block to model the resistive and delay characteristics of a relay controlled by an external physical signal.

New Fuse Block

The Passive Devices library now contains a Fuse block to model the following fuse characteristics:

- Resistance.
- Rated current at which the fuse blows when exceeded for a specified amount of time.

New NMOS and PMOS Blocks

The SPICE-Compatible Semiconductors library (in the Semiconductor Devices library) now contains NMOS and PMOS blocks to model the electrical characteristics of SPICE Level-1 MOSFET devices.

R2008a+

Version: 1.0

New Features

Product Introduction

SimElectronics software is a modeling environment for the engineering design and simulation of electronic and electromechanical systems within the Simulink® environment.

Version 1.0 includes these features:

- A library of electronic and electromechanical blocks that model components such as:
 - Sensors
 - Semiconductors
 - Actuators

For these blocks, you enter key parameter values directly from industry datasheets.

For more information about the available blocks, see [SimElectronics Block Libraries](#).

- A function, `netlist2sl`, for creating library blocks that represent circuit data in a SPICE netlist.
- Ability to convert SimElectronics models to C code.

For more information about code generation, see [Code Generation in the Simscape documentation](#).

- Access to linearization and steady-state solve capabilities in Simscape.

For more information about linearization, see [Linearizing at an Operating Point in the Simscape documentation](#).

For more information about how Simscape solves models, see [How Simscape Simulation Works in the Simscape documentation](#).